

Copyright  
by  
Kyle Jordan DeMars  
2010

The Dissertation Committee for Kyle Jordan DeMars  
certifies that this is the approved version of the following dissertation:

**Nonlinear Orbit Uncertainty Prediction and Rectification for  
Space Situational Awareness**

Committee:

---

Robert H. Bishop, Supervisor

---

Maruthi R. Akella

---

Belinda G. Marchand

---

Timothy P. Crain

---

Moriba K. Jah

**Nonlinear Orbit Uncertainty Prediction and Rectification for  
Space Situational Awareness**

**by**

**Kyle Jordan DeMars, B.S.As.E., M.S.E.**

**DISSERTATION**

Presented to the Faculty of the Graduate School of  
The University of Texas at Austin  
in Partial Fulfillment  
of the Requirements  
for the Degree of

**DOCTOR OF PHILOSOPHY**

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2010

In memory of my grandmother, Martha Irene Murry.

## Acknowledgments

First, I would like to express my sincere gratitude to my advisor, Dr. Robert H. Bishop, who has not only allowed me a great deal of freedom in the exploration of this dissertation topic, but has also provided guidance and direction when it was needed. It has truly been an honor to have him as my advisor and mentor during my time at the university.

I would also like to extend this appreciation to my committee members, Dr. Maruthi R. Akella, Dr. Belinda G. Marchand, Dr. Timothy P. Crain of the NASA Johnson Space Center, and Dr. Moriba K. Jah of the Air Force Research Laboratory for their assistance in the preparation of this dissertation. I would like to especially thank Dr. Jah for our discussions on almost all aspects of this dissertation.

No acknowledgments would be complete without a thank you to all of my colleagues at The University of Texas, of whom there are too many to list. However, I would like to especially thank Dr. Renato Zanetti, Dr. Eduardo Gildin, Jorge Alvarez, Tyler Summers, Travis Mercker, and Hector Escobar for all of the discussions and distractions each has provided in their own way.

And last, but certainly not least, I wish to thank my family. Their unending support has incalculably helped me to get to this point and has been absolutely essential throughout my studies and the process of preparing this dissertation.

# Nonlinear Orbit Uncertainty Prediction and Rectification for Space Situational Awareness

Publication No. \_\_\_\_\_

Kyle Jordan DeMars, Ph.D.  
The University of Texas at Austin, 2010

Supervisor: Robert H. Bishop

A new method for predicting the uncertainty in a nonlinear dynamical system is developed and analyzed in the context of uncertainty evolution for resident space objects (RSOs) in the near-geosynchronous orbit regime under the influence of central body gravitational acceleration, third body perturbations, and attitude-dependent solar radiation pressure (SRP) accelerations and torques. The new method, termed the splitting Gaussian mixture unscented Kalman filter (SGMUKF), exploits properties of the differential entropy or Rényi entropy for a linearized dynamical system to determine when a higher-order prediction of uncertainty reaches a level of disagreement with a first-order prediction, and then applies a multivariate Gaussian splitting algorithm to reduce the impact of induced nonlinearity. In order to address the relative accuracy of the new method with respect to the more traditional approaches of the extended Kalman filter (EKF) and unscented Kalman filter (UKF), several concepts regarding the comparison of probability density functions (pdfs) are introduced and utilized in the analysis.

The research also describes high-fidelity modeling of the nonlinear dynamical system which drives the motion of an RSO, and includes models for evaluation of the central body gravitational acceleration, the gravitational acceleration due to other celestial bodies, and attitude-dependent SRP accelerations and torques when employing a macro plate model of an RSO. Furthermore, a high-fidelity model of the measurement of the line-of-sight of a spacecraft from a ground station is presented, which applies light-time and stellar aberration corrections, and accounts for observer and target lighting conditions, as well as for the sensor field of view.

The developed algorithms are applied to the problem of forward predicting the time evolution of the region of uncertainty for RSO tracking, and uncertainty rectification via the fusion of incoming measurement data with prior knowledge. It is demonstrated that the SGMUKF method is significantly better able to forward predict the region of uncertainty and is subsequently better able to utilize new measurement data.

# Table of Contents

<b>Acknowledgments</b>	<b>v</b>
<b>Abstract</b>	<b>vi</b>
<b>List of Tables</b>	<b>xii</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Acronyms</b>	<b>xvi</b>
<b>Chapter 1. Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Review of Existing Literature . . . . .	2
1.3 Research Contributions . . . . .	6
1.4 Organization of the Dissertation . . . . .	7
<b>Chapter 2. Concepts in Probability</b>	<b>10</b>
2.1 The Gaussian and Gaussian Mixture Model Distributions . . . . .	10
2.2 Probability Density Function Measures . . . . .	12
2.2.1 Distance Measure between Distributions . . . . .	12
2.2.1.1 $L_2$ Distance between GMMs . . . . .	14
2.2.1.2 Normalized $L_2$ Distance between GMMs . . . . .	16
2.2.2 Likelihood Agreement between Distributions . . . . .	17
2.2.3 Differential Entropy and Rényi Entropy of a Gaussian Distribution	19
2.3 Splitting a Gaussian Distribution . . . . .	22
2.3.1 The Univariate Case . . . . .	23
2.3.2 The Multivariate Case . . . . .	28
2.4 Merging Gaussian Distributions . . . . .	31
2.4.1 Method of Moments . . . . .	31
2.4.2 Top-Down Merging Method . . . . .	34



<b>Chapter 3. Recursive Filtering Strategies</b>	<b>36</b>
3.1 The Extended Kalman Filter . . . . .	37
3.1.1 Propagation . . . . .	37
3.1.2 Update . . . . .	39
3.2 The Unscented Kalman Filter . . . . .	41
3.2.1 The Unscented Transform . . . . .	43
3.2.2 Propagation . . . . .	45
3.2.3 Update . . . . .	46
3.3 The Gaussian Mixture Extended Kalman Filter . . . . .	47
3.3.1 Propagation . . . . .	48
3.3.2 Update . . . . .	50
3.4 The Gaussian Mixture Unscented Kalman Filter . . . . .	53
3.4.1 Propagation . . . . .	54
3.4.2 Update . . . . .	56
3.5 The Splitting Gaussian Mixture Unscented Kalman Filter . . . . .	59
3.5.1 Detecting Nonlinearity during Propagation . . . . .	59
3.5.2 Propagation . . . . .	66
3.5.3 Update . . . . .	68
3.6 Summary of Filter Algorithms . . . . .	70
3.6.1 EKF Algorithm . . . . .	70
3.6.2 UKF Algorithm . . . . .	72
3.6.3 GMEKF Algorithm . . . . .	74
3.6.4 GMUKF Algorithm . . . . .	75
3.6.5 SGMUKF Algorithm . . . . .	77
<b>Chapter 4. Dynamics and Measurement Modeling</b>	<b>81</b>
4.1 Dynamics Modeling . . . . .	81
4.1.1 Central Body Gravitational Acceleration . . . . .	85
4.1.1.1 Point Mass Gravitational Acceleration . . . . .	86
4.1.1.2 Zonal Harmonics Gravitational Acceleration . . . . .	87
4.1.1.3 Spherical Harmonics Gravitational Acceleration . . . . .	92
4.1.2 Third-Body Gravitational Acceleration . . . . .	100

4.1.3	Solar Radiation Pressure Acceleration . . . . .	101
4.1.4	Solar Radiation Pressure Moment . . . . .	104
4.1.5	Models for the Shadow Factor . . . . .	107
4.1.5.1	Cylindrical Model . . . . .	107
4.1.5.2	Conic Model . . . . .	108
4.2	Measurement Modeling . . . . .	110
4.2.1	Light Time Correction . . . . .	110
4.2.2	Stellar Aberration Correction . . . . .	112
4.2.3	Lighting Conditions . . . . .	113
4.2.3.1	Satellite Lighting Condition . . . . .	114
4.2.3.2	Observer Lighting Condition . . . . .	114
4.2.4	Field of View Condition . . . . .	115
4.2.5	Right Ascension and Declination Measurements . . . . .	116
<b>Chapter 5.</b>	<b>Results</b>	<b>120</b>
5.1	Propagation in a Simplified Tracking Model . . . . .	121
5.1.1	Circular Orbit Test Case . . . . .	125
5.1.2	Eccentric Orbit Test Case . . . . .	135
5.2	Propagation in the Full Tracking Model . . . . .	145
5.3	Update in the Full Tracking Model . . . . .	167
<b>Chapter 6.</b>	<b>Conclusions</b>	<b>177</b>
6.1	Research Summary . . . . .	177
6.2	Future Research Considerations . . . . .	178
<b>Appendices</b>		<b>180</b>
<b>Appendix A.</b>	<b>Attitude Considerations</b>	<b>181</b>
A.1	Averaging Quaternions . . . . .	182
A.2	The Gaussian and Gaussian Mixture Model Distributions . . . . .	184
A.3	Splitting a Multivariate Gaussian Distribution . . . . .	186
A.4	The Method of Moments . . . . .	189
A.5	The Kalman Filter Update . . . . .	191
A.6	The Unscented Transform . . . . .	193

<b>Bibliography</b>	<b>198</b>
<b>Vita</b>	<b>209</b>

## List of Tables

2.1	Parameters and Performance of Splitting Libraries . . . . .	25
2.2	3-Component Splitting Library . . . . .	25
2.3	4-Component Splitting Library . . . . .	26
2.4	5-Component Splitting Library . . . . .	26
4.1	Legendre Polynomials and their Derivatives . . . . .	88
5.1	Magnitude of Typical Spacecraft Accelerations <sup>9</sup> . . . . .	121
5.2	Plate Property Specification for the Hexagonal Prism Model . . . . .	152

## List of Figures

2.1	Components of the splitting libraries and their sum as compared to the normal Gaussian distribution for $k_2 = 3$ . . . . .	27
2.2	Components of the splitting libraries and their sum as compared to the normal Gaussian distribution for $k_2 = 4$ . . . . .	27
2.3	Components of the splitting libraries and their sum as compared to the normal Gaussian distribution for $k_2 = 5$ . . . . .	28
4.1	Cylindrical Shadow Model . . . . .	108
4.2	Geometry of the Penumbra Phase . . . . .	109
5.1	Nominal trajectories for the circular (in blue) and eccentric (in red) test cases in the simplified tracking model. . . . .	124
5.2	Likelihood agreement measure for the UKF and SGMUKF, normalized by the value for the SGMUKF . . . . .	126
5.3	Position pdf contours with monte carlo samples at epoch. . . . .	128
5.4	Velocity pdf contours with monte carlo samples at epoch. . . . .	129
5.5	Position pdf contours with monte carlo samples at one period of the nominal orbit. . . . .	130
5.6	Velocity pdf contours with monte carlo samples at one period of the nominal orbit. . . . .	131
5.7	Position pdf contours with monte carlo samples at two periods of the nominal orbit. . . . .	132
5.8	Velocity pdf contours with monte carlo samples at two periods of the nominal orbit. . . . .	133
5.9	Position pdf contours with monte carlo samples at three periods of the nominal orbit. . . . .	134
5.10	Velocity pdf contours with monte carlo samples at three periods of the nominal orbit. . . . .	135
5.11	Likelihood agreement measure for the UKF and SGMUKF, normalized by the value for the SGMUKF . . . . .	136
5.12	Position pdf contours with monte carlo samples at epoch. . . . .	138

5.13	Velocity pdf contours with monte carlo samples at epoch. . . . .	139
5.14	Position pdf contours with monte carlo samples at one period of the nominal orbit. . . . .	140
5.15	Velocity pdf contours with monte carlo samples at one period of the nominal orbit. . . . .	141
5.16	Position pdf contours with monte carlo samples at two periods of the nominal orbit. . . . .	142
5.17	Velocity pdf contours with monte carlo samples at two periods of the nominal orbit. . . . .	143
5.18	Position pdf contours with monte carlo samples at three periods of the nominal orbit. . . . .	144
5.19	Velocity pdf contours with monte carlo samples at three periods of the nominal orbit. . . . .	145
5.20	Nominal trajectory for the full tracking model. . . . .	149
5.21	Hexagonal prism flat plate model, adapted from [54]. . . . .	151
5.22	Likelihood agreement measure for the UKF, 3-component SGMUKF, and 5-component SGMUKF, normalized by the value for the 5-component SGMUKF . . . . .	154
5.23	Position ( $x - y$ projection) pdf contours with monte carlo samples at one orbit period. . . . .	156
5.24	Position ( $x - z$ projection) pdf contours with monte carlo samples at one orbit period. . . . .	157
5.25	Position ( $y - z$ projection) pdf contours with monte carlo samples at one orbit period. . . . .	158
5.26	Velocity ( $x - y$ projection) pdf contours with monte carlo samples at one orbit period. . . . .	159
5.27	Velocity ( $x - z$ projection) pdf contours with monte carlo samples at one orbit period. . . . .	160
5.28	Velocity ( $y - z$ projection) pdf contours with monte carlo samples at one orbit period. . . . .	161
5.29	Attitude (roll-pitch projection) pdf contours with monte carlo samples at one orbit period. . . . .	162
5.30	Attitude (roll-yaw projection) pdf contours with monte carlo samples at one orbit period. . . . .	163
5.31	Attitude (pitch-yaw projection) pdf contours with monte carlo samples at one orbit period. . . . .	164
5.32	Angular Velocity (body-frame $x - y$ projection) pdf contours with monte carlo samples at one orbit period. . . . .	165

5.33	Angular Velocity (body-frame $x - z$ projection) pdf contours with monte carlo samples at one orbit period. . . . .	166
5.34	Angular Velocity (body-frame $y - z$ projection) pdf contours with monte carlo samples at one orbit period. . . . .	167
5.35	Selected point for study of the update in the full tracking model. . . .	169
5.36	Projected position pdf surfaces with true state before any updates. . .	172
5.37	Projected position pdf surfaces with true state after one update. . . .	173
5.38	Projected position pdf surfaces with true state after two updates. . . .	174
5.39	Projected position pdf surfaces with true state after ten updates. . . .	175
5.40	Projected position pdf surfaces with true state after all updates. . . .	176

## List of Acronyms

<b>AFRL</b>	Air Force Research Laboratory
<b>CDKF</b>	central difference Kalman filter
<b>DMM</b>	Dirac mixture model
<b>DPD</b>	density power divergence
<b>EKF</b>	extended Kalman filter
<b>GMEKF</b>	Gaussian mixture extended Kalman filter
<b>GMM</b>	Gaussian mixture model
<b>GМУKF</b>	Gaussian mixture unscented Kalman filter
<b>NRL</b>	Naval Research Laboratory
<b>pdf</b>	probability density function
<b>RSO</b>	resident space object
<b>SGMUKF</b>	splitting Gaussian mixture unscented Kalman filter
<b>SPKF</b>	sigma-point Kalman filter
<b>SRP</b>	solar radiation pressure



**UKF**      unscented Kalman filter

**UT**      unscented transform

# Chapter 1

## Introduction

### 1.1 Motivation

Beginning with the launch of Sputnik I in October 1957, a catalog of space objects has been maintained, first by the Naval Research Laboratory (NRL) and subsequently by the Air Force Research Laboratory (AFRL) via a transfer of the program from NRL to AFRL during 2003-2004.<sup>56</sup> Since the launch of Sputnik I, the number of objects in orbit coming from new launches, decommissioned satellites, and debris created by collision of objects in orbit has posed an ever increasing challenge to the cataloging of space objects. As of 2006, there were approximately 9000 space objects being tracked by the U.S. Space Surveillance Network and maintained in the satellite catalog.<sup>38</sup> This number will inevitably increase as more objects are launched and as more collisions occur.

The proliferation of resident space objects (RSOs), both inactive and active, leads to interest not only in the avoidance of collisions between objects, but also in determining the intent of specific objects. However, since specific objects are not constantly monitored, information relating to the current location/orientation of the objects is limited. Limited availability of tracking data ultimately requires long-term predictions of where objects will be in order to improve the probability of reacquiring

the object with followup tracking. Since the space environment is nonlinear by nature and since uncertainty is inevitably present in the state (position, velocity, etc.) of an RSO, long-term accurate prediction of the state and uncertainty of RSOs is a nontrivial task.

## 1.2 Review of Existing Literature

The rapid advances in existing recursive algorithms which enable the tracking of RSOs originate with Kalman's seminal paper on a state-space approach to stochastic estimation via what is now known as the Kalman filter.<sup>28</sup> In his paper, Kalman outlined the general approach which permeates throughout the majority of the estimation algorithms utilized from that time forward. This approach consists of a two-step procedure, comprised firstly by a propagation stage in which the state of a dynamical system along with its uncertainty are projected forward in time, and secondly by an update stage in which new information that is made available via incomplete and imperfect measurements of the state are utilized in such a way so as to rectify the state and reduce the uncertainty if possible. The two-step procedure is then repeated, making use of the measurements whenever they become available. This repetition therefore establishes the recursive nature of the overall algorithm.

Introduction of the Kalman filter into the literature spawned rapid advances in the applicability and implementation of recursive estimation to dynamical systems. The first, and arguably most influential, advance in applicability was the introduction of the extended Kalman filter (EKF) in Smith's report.<sup>59</sup> The EKF established an approach to the estimation of nonlinear dynamical systems by proposing linearization

of the dynamical system and observational relationships about the current best estimated state. The EKF was then subsequently applied to the problem of estimation for space-borne platforms and ultimately to the ability for landing humans on the Moon. However, the EKF is not without limitations. The utilization of linearization in order to accomplish the recursive filtering limits the range of applications to those in which the linearization holds with respect to the time-scale of the observations. That is, if a nonlinear dynamical system is accurately described by a first-order linearization approach over the time span between consecutive measurements, then the EKF can be utilized to provide accurate estimates of the system's state as well as the uncertainty present in the state. To address the cases in which linearization of the nonlinear dynamical system does not accurately reflect the nonlinear behavior and higher-order effects begin to play a role, the second-order EKF scheme was described by Athans,<sup>4</sup> and has been shown to yield improvements (especially when considering the observational relationships) when applied to orbit estimation problems.<sup>20,73,75</sup> While the second-order EKF does, as its names indicates, include the second-order terms of a Taylor series expansion in both the propagation and update stages of the Kalman filter structure, the governing equations for the second-order EKF are based on the assumption of normality of the state errors, which limits the applicability. More recently, a class of so-called sigma-point Kalman filters (SPKFs) has emerged, chief among them the unscented Kalman filter (UKF)<sup>25,26</sup> and the central difference Kalman filter (CDKF).<sup>21,70</sup> The UKF is based on the proposition that the distribution of a state is easier to approximate than it is to consider arbitrarily high order terms in a Taylor series expansion of nonlinear equations.<sup>67</sup> In both the EKF and

UKF approaches to recursive estimation, the implicit assumption is that the uncertainty present in the state of the dynamical system is well represented by only the first two statistical moments (i.e. the mean and covariance) of the distribution.

To relax the necessity of assuming that the first two statistical moments are sufficient for an accurate description of the uncertainty, Sorenson introduced a Gaussian mixture model (GMM) approach to the Bayesian estimation problem which allows for the modeling of the distribution by a sum of Gaussian component distributions and the application of parallel-operating filters.<sup>61</sup> Alspach then utilized the GMM approach of Sorenson to describe a nonlinear recursive estimation scheme in which each component of the GMM distribution is filtered via an EKF.<sup>1</sup> Subsequently, the approach taken in Sorenson to apply the EKF to the components of the GMM was modified so as to apply the UKF methodology to the components of the GMM. Therefore, the same Kalman filtering paradigm can be applied to a nonlinear dynamical system in which the total state uncertainty description is not well represented by only the first two statistical moments. Recently, the Gaussian mixture approach has been extended by Terejanu to adapt the GMM component weights during propagation of the GMM probability density function (pdf)<sup>66</sup> and applied to the orbit determination problem in the presence of solar radiation pressure (SRP) effects<sup>11</sup> and drag effects,<sup>12</sup> both of which show improvements in the tracking of RSOs when compared to implementation of the UKF.

Another class of recursive estimation strategies which circumvent the assumption that the first two statistical moments are sufficient is based on higher-order moments and utilizing these moments during the propagation and update stages of the

filter. For instance, Park has described a method for utilizing state transition tensors, which are higher-order state transition matrices, in order to nonlinearly map statistics through the nonlinear dynamical system and observational relationships.<sup>48,49</sup> Building upon Park's work, Majji derives differential equations for the statistical moments of the state uncertainty beyond the first and second-order.<sup>41</sup> In both cases, implementation of the higher-order methods requires the computation of higher-order derivatives to implement the Taylor series expansions. This requires either an analytic formulation of the higher-order derivatives of nonlinear equations (which can become quite cumbersome even for relatively benign nonlinear equations) or the implementation of numerical procedures such as finite differences,<sup>51</sup> automatic differentiation,<sup>15</sup> or complex-step derivatives<sup>62</sup> to determine the requisite derivatives. In the case of the latter option, Lai has implemented a complex-step derivative approach as an alternative method for applying second-order filters.<sup>32,33</sup> Recently, Lantoine has reported a method for numerically computing higher-order derivatives based on the complex-step derivative technique, which could be directly applied to the approaches of Park and Majji. However, the inclusion of higher-order terms naturally requires significantly more computational power and, as is the case of the complex-step derivative method, the implementation of operator overloading in software. The advantage of the higher-order methods is illustrated by Park<sup>49</sup> wherein the implementation of state transition tensors is better able to accurately account for nonlinearities in the transformation of the statistical moments in order to better describe the uncertainty for spacecraft navigation.

### 1.3 Research Contributions

This work focuses on recursive methods which describe the state uncertainty for the orbit determination problem. The main contribution of the dissertation is the development of a new technique which makes use of a method for the detection of nonlinearity during the time prediction of state uncertainty and subsequently utilizes a splitting technique to decrease the errors made by low-order Taylor series approximations of the nonlinear system. This new method is shown to be able to better approximate the propagation of uncertainty through a nonlinear dynamical system than standard approaches are capable of doing. The standard approaches rely on first and second-order approximations in order to predict the uncertainty along a nominal path; in doing so, these methods do not allow for cases in which the volume of uncertainty becomes large enough that the low-order approximations are no longer valid. Furthermore, the new method does not require the implementation of higher-order schemes, such as that of Park<sup>48,49</sup> or Majji,<sup>41</sup> thereby avoiding the numerical and computational difficulties involved with higher-order methods.

To develop the technique of splitting, several other contributions are encountered, such as developing more generalized algorithms for the splitting and merging of GMM distributions than currently exist in the literature. While the presented method for splitting a distribution is not altogether new, a new approach which relies on numerical minimization of closed-form measures of distance between pdfs is presented and applied. Finally, as with all recursive filtering schemes, modeling of the dynamical and observational relationships plays a central role. To this extent, a thorough treatment of the gravitational acceleration imparted on a spacecraft by a

central body is given, including a non-singular implementation for the evaluation of high-fidelity spherical harmonics models. Also of interest in the dynamical systems modeling is that of the SRP, which imparts both an acceleration and a moment. Again, the core algorithms are extended from those found in the existing literature to yield not only an acceleration due to SRP, but also a moment due to SRP, which directly influences the evolution of the orientation of the vehicle. Each of these improvements in modeling is then applied to the new filtering scheme that is developed based on splitting techniques.

## 1.4 Organization of the Dissertation

The organization of the dissertation is as follows: in Chapter 2, several concepts relating to pdfs are presented. This discussion begins with the general specification of the Gaussian distribution via its pdf and the GMM distribution via its pdf in Section 2.1. In Section 2.2 several measures for pdfs are presented, such as the  $L_2$  and the normalized  $L_2$  distances between two pdfs which are characterized by GMMs. Furthermore, a likelihood measure for determining how likely a GMM pdf is with respect to a set of sample points is presented, and the average amount of surprisal present in a Gaussian distribution is presented. Section 2.3 then utilizes the  $L_2$  distance measure to develop a method for splitting a Gaussian distribution into smaller Gaussian distributions, such that a single Gaussian distribution can be approximated via a GMM. This approach to approximating a Gaussian distribution by a GMM is first presented for a univariate Gaussian distribution and then extended to the multivariate case. Finally, Section 2.4 presents a method for taking a GMM



distribution and reducing the number of components to form a reduced-component GMM via merging components of the original GMM.

Chapter 3 details the development of the recursive filtering strategies considered in this work. The EKF, which serves as the fundamental filtering scheme for most all nonlinear systems is derived in Section 3.1. Additionally, the UKF, which is a higher-order extension of the EKF, is described in Section 3.2. Sections 3.3 and 3.4 apply the strategies of the EKF and UKF, respectively, to the situation in which a GMM is used to describe the uncertainty in the state. Finally, building upon the prior developments of the chapter, Section 3.5 develops a splitting Gaussian mixtures recursive filtering scheme in order to describe a method which attempts to approximate the forward evolution of uncertainty by online adaptation of the GMM which describes the state uncertainty.

Chapter 4 presents the dynamical and observational relationships which are necessary for the implementation of the recursive filtering strategies outlined in Chapter 3. For the dynamics, the application of gravitational acceleration induced by a central body is introduced in Section 4.1.1. The models presented for the central body acceleration include: the point mass model in Section 4.1.1.1, the zonal harmonics model in Section 4.1.1.2, and the spherical harmonics model in Section 4.1.1.3. Additionally, the inclusion of gravitational acceleration caused by the presence of gravitating bodies other than the central body is discussed in Section 4.1.2. The final dynamical model is that of the SRP, for which the acceleration model is given in Section 4.1.3, and the moment model is given in Section 4.1.4. Next, the observational relationships are given in Section 4.2. The observational relationship considered in

this work is that of a line-of-sight given by angular observations, which is described in Section 4.2.5. In conjunction with the angular observations, methods which account for corrections to the measurements and conditions upon which measurements may be made are described in Sections 4.2.1–4.2.4.

Chapter 5 presents the analysis and performance comparison of the developed recursive filtering algorithms in the context of orbit uncertainty prediction and rectification. The analysis is broken into three parts, each building upon the previous: orbit uncertainty prediction using lower fidelity dynamics models, orbit uncertainty prediction in using high fidelity dynamics models, and orbit uncertainty rectification via processing of measurement data.

A summary of the results, as well as some conclusions and future directions of research are given in Chapter 6.

## Chapter 2

### Concepts in Probability

#### 2.1 The Gaussian and Gaussian Mixture Model Distributions

Given a continuous random vector,  $\mathbf{x} \in \mathbb{R}^n$ , the probability density function (pdf) is a function which describes the relative likelihood of the random variable across points in  $\mathbb{R}^n$ . The pdf is a nonnegative function which when integrated over its entire support set is one. The most widely used pdf is the Gaussian pdf. Let  $\mathbf{x}$  be a Gaussian random variable of dimension  $n$ , with mean and covariance denoted by  $\mathbf{m} \in \mathbb{R}^n$  and  $\mathbf{P} = \mathbf{P}^T > \mathbf{0} \in \mathbb{R}^{n \times n}$ , respectively. Then, the pdf for  $\mathbf{x}$  is defined as

$$p_g(\mathbf{x}; \mathbf{m}, \mathbf{P}) = |2\pi\mathbf{P}|^{-1/2} \exp \left\{ -\frac{1}{2}(\mathbf{x} - \mathbf{m})^T \mathbf{P}^{-1}(\mathbf{x} - \mathbf{m}) \right\}, \quad (2.1)$$

where  $|\cdot|$  represents the matrix determinant. The ubiquitous use of the Gaussian pdf is motivated by its ability to model random events, such as noise. As is seen in Eq. (2.1), the pdf is completely characterized by the mean and covariance, which leads to the important property that the moments of a Gaussian random variable can be written in terms of only the mean and the covariance. If two Gaussian pdfs are multiplied together, the resultant product is a scaled Gaussian pdf given by<sup>44</sup>

$$p_g(\mathbf{x}; \mathbf{m}_1, \mathbf{P}_1)p_g(\mathbf{x}; \mathbf{m}_2, \mathbf{P}_2) = K(\mathbf{m}_1, \mathbf{m}_2, \mathbf{P}_1, \mathbf{P}_2)p_g(\mathbf{x}; \mathbf{m}_3, \mathbf{P}_3) \quad (2.2)$$

where

$$\begin{aligned}
\mathbf{m}_3 &= \mathbf{P}_3 (\mathbf{P}_1^{-1} \mathbf{m}_1 + \mathbf{P}_2^{-1} \mathbf{m}_2) \\
\mathbf{P}_3 &= (\mathbf{P}_1^{-1} + \mathbf{P}_2^{-1})^{-1} \\
K(\mathbf{m}_1, \mathbf{m}_2, \mathbf{P}_1, \mathbf{P}_2) &= |2\pi(\mathbf{P}_1 + \mathbf{P}_2)|^{-1/2} \\
&\times \exp \left\{ -\frac{1}{2}(\mathbf{m}_1 - \mathbf{m}_2)^T (\mathbf{P}_1 + \mathbf{P}_2)^{-1} (\mathbf{m}_1 - \mathbf{m}_2) \right\}.
\end{aligned} \tag{2.3}$$

Eq. (2.2) serves to establish the important property of the Gaussian pdf (which will be utilized in the sequel) that the integral of the product of two Gaussian pdfs is given by

$$\int_{\mathbb{R}^n} p_g(\mathbf{x}; \mathbf{m}_1, \mathbf{P}_1) p_g(\mathbf{x}; \mathbf{m}_2, \mathbf{P}_2) d\mathbf{x} = K(\mathbf{m}_1, \mathbf{m}_2, \mathbf{P}_1, \mathbf{P}_2) \tag{2.4}$$

where  $K(\cdot)$  is given in Eq. (2.3).

A direct extension of the Gaussian pdf is the so-called Gaussian mixture pdf, or Gaussian mixture model (GMM), which is given by a sum of weighted Gaussian pdfs, i.e.

$$p(\mathbf{x}) = \sum_{i=1}^L w_i p_g(\mathbf{x}; \mathbf{m}_i, \mathbf{P}_i). \tag{2.5}$$

In Eq. (2.5),  $L$  represents the number of components of the GMM,  $w_i$  are the weights associated with each component,  $\mathbf{m}_i$  are the means associated with each component, and  $\mathbf{P}_i$  are the covariances associated with each component. To retain the properties of a valid pdf (that is, to ensure positivity across the support of the pdf and to ensure that the area under the pdf is one), the weights must all be positive and must sum

to one, that is

$$w_i \geq 0 \quad \forall \quad i \in \{1, 2, \dots, L\} \quad \text{and} \quad \sum_{i=1}^L w_i = 1.$$

The GMM approach to describing a pdf retains the benefits of the easy characterization and interpretation of the Gaussian pdf, while simultaneously extending the applicability of the Gaussian pdf since a large class of pdfs can be approximated using the GMM approach. This point was illustrated by Sorenson<sup>61</sup> where it was shown that the approximation of a pdf by a GMM pdf converges uniformly as the number of components in the GMM approximation increases without bound. This is a highly intuitive result since each component of the GMM approaches an impulse function as the component covariance decreases to zero. Therefore, by decreasing the component covariance, increasing the number of components, and distributing the component means properly, one can readily approximate the shape of a large class of pdfs.

## 2.2 Probability Density Function Measures

### 2.2.1 Distance Measure between Distributions

To develop a measure of distance between two pdfs, first consider the density power divergence (DPD) given by<sup>6</sup>

$$\text{DPD}_\alpha(p_1, p_2) = \int_{\mathbb{R}^n} \left\{ p_2^{1+\alpha}(\mathbf{x}) - \left(1 + \frac{1}{\alpha}\right) p_1(\mathbf{x}) p_2^\alpha(\mathbf{x}) + \frac{1}{\alpha} p_1^{1+\alpha}(\mathbf{x}) \right\} d\mathbf{x},$$

where  $\alpha$  is the control parameter of the DPD and  $p_1(\mathbf{x})$  and  $p_2(\mathbf{x})$  are the input pdfs for which the DPD is computed. Furthermore, it is noted that for  $\alpha = 0$ , the integrand is undefined, and therefore  $\text{DPD}_0(p_1, p_2)$  is defined to be the Kullback-

Leibler divergence<sup>6,31</sup>

$$\text{KL}(p_1, p_2) = \lim_{\alpha \rightarrow 0} d_\alpha(p_1, p_2) = \int_{\mathbb{R}^n} p_1(\mathbf{x}) \log \frac{p_1(\mathbf{x})}{p_2(\mathbf{x})} d\mathbf{x} .$$

In the case where  $\alpha = 1$ , it is seen that the DPD becomes

$$\text{L}_2(p_1, p_2) = \int_{\mathbb{R}^n} (p_1(\mathbf{x}) - p_2(\mathbf{x}))^2 d\mathbf{x} ,$$

which is simply the  $\text{L}_2$  distance between the surfaces described by the two pdfs. Therefore the control parameter  $\alpha$  gives a smooth bridge between the Kullback-Leibler divergence and the  $\text{L}_2$  distance for  $0 < \alpha < 1$ .<sup>23,24</sup> It is worth noting that while the Kullback-Leibler divergence is not symmetric (i.e.  $\text{KL}(p_1, p_2) \neq \text{KL}(p_2, p_1)$ ) and does not satisfy the triangle inequality, it is readily observed that the  $\text{L}_2$  distance satisfies both of these properties. Symmetry is guaranteed by the square in the integrand, and satisfaction of the triangle inequality can be shown by observing that

$$\begin{aligned} \text{L}_2(p_1, p_2) &= \int_{\mathbb{R}^n} p_1^2(\mathbf{x}) d\mathbf{x} + \int_{\mathbb{R}^n} p_2^2(\mathbf{x}) d\mathbf{x} - 2 \int_{\mathbb{R}^n} p_1(\mathbf{x}) p_2(\mathbf{x}) d\mathbf{x} \\ &= \text{L}_2(p_1) + \text{L}_2(p_2) - 2 \int_{\mathbb{R}^n} p_1(\mathbf{x}) p_2(\mathbf{x}) d\mathbf{x} . \end{aligned} \quad (2.6)$$

Since  $p_1(\mathbf{x})$  and  $p_2(\mathbf{x})$  are both non-negative over the domain of integration,

$$\int_{\mathbb{R}^n} p_1(\mathbf{x}) p_2(\mathbf{x}) d\mathbf{x} \geq 0 , \quad (2.7)$$

which, by substituting Eq. (2.7) back into Eq. (2.6), then gives the triangle inequality as

$$\text{L}_2(p_1, p_2) \leq \text{L}_2(p_1) + \text{L}_2(p_2) .$$

### 2.2.1.1 $L_2$ Distance between GMMs

Now, consider the  $L_2$  distance between two GMM pdfs. Recall that the  $L_2$  distance is defined as

$$L_2(p_1, p_2) = \int_{\mathbb{R}^n} (p_1(\mathbf{x}) - p_2(\mathbf{x}))^2 d\mathbf{x},$$

such that, upon expanding terms, we have the accumulation of three terms

$$L_2(p_1, p_2) = \int_{\mathbb{R}^n} p_1^2(\mathbf{x}) d\mathbf{x} + \int_{\mathbb{R}^n} p_2^2(\mathbf{x}) d\mathbf{x} - 2 \int_{\mathbb{R}^n} p_1(\mathbf{x}) p_2(\mathbf{x}) d\mathbf{x}. \quad (2.8)$$

Assume that the pdfs  $p_1(\mathbf{x})$  and  $p_2(\mathbf{x})$  are given by GMMs of the form

$$p_1(\mathbf{x}) = \sum_{i=1}^{k_1} w_{1,i} p_g(\mathbf{x}; \mathbf{m}_{1,i}, \mathbf{P}_{1,i}) \quad (2.9a)$$

$$p_2(\mathbf{x}) = \sum_{j=1}^{k_2} w_{2,j} p_g(\mathbf{x}; \mathbf{m}_{2,j}, \mathbf{P}_{2,j}). \quad (2.9b)$$

Since all three terms in Eq. (2.8) are of the same form, that is they all appear as integrals of the product of two GMMs up to scaling, the third term can be considered and the results can be extended to the first two terms. Looking at only the final term in Eq. (2.8) and substituting Eqs. (2.9) in place of the terms in the integrand yields

$$\begin{aligned} \int_{\mathbb{R}^n} p_1(\mathbf{x}) p_2(\mathbf{x}) d\mathbf{x} &= \int_{\mathbb{R}^n} \left[ \sum_{i=1}^{k_1} w_{1,i} p_g(\mathbf{x}; \mathbf{m}_{1,i}, \mathbf{P}_{1,i}) \right] \left[ \sum_{j=1}^{k_2} w_{2,j} p_g(\mathbf{x}; \mathbf{m}_{2,j}, \mathbf{P}_{2,j}) \right] d\mathbf{x} \\ &= \int_{\mathbb{R}^n} \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} w_{1,i} w_{2,j} p_g(\mathbf{x}; \mathbf{m}_{1,i}, \mathbf{P}_{1,i}) p_g(\mathbf{x}; \mathbf{m}_{2,j}, \mathbf{P}_{2,j}) d\mathbf{x} \\ &= \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} w_{1,i} w_{2,j} \int_{\mathbb{R}^n} p_g(\mathbf{x}; \mathbf{m}_{1,i}, \mathbf{P}_{1,i}) p_g(\mathbf{x}; \mathbf{m}_{2,j}, \mathbf{P}_{2,j}) d\mathbf{x} \\ &= \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} w_{1,i} w_{2,j} K(\mathbf{m}_{1,i}, \mathbf{m}_{2,j}, \mathbf{P}_{1,i}, \mathbf{P}_{2,j}). \end{aligned} \quad (2.10)$$

In the preceding development, the integral of the sum was replaced by the sum of the integral in going from the second step to the third. This replacement is possible due to the linearity of the integration operator. Similarly, by applying the same process as used in developing Eq. (2.10), the first two terms in Eq. (2.8) are given by

$$\int_{\mathbb{R}^n} p_1^2(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^{k_1} \sum_{j=1}^{k_1} w_{1,i} w_{1,j} K(\mathbf{m}_{1,i}, \mathbf{m}_{1,j}, \mathbf{P}_{1,i}, \mathbf{P}_{1,j}) \quad (2.11)$$

and

$$\int_{\mathbb{R}^n} p_2^2(\mathbf{x}) d\mathbf{x} = \sum_{i=1}^{k_2} \sum_{j=1}^{k_2} w_{2,i} w_{2,j} K(\mathbf{m}_{2,i}, \mathbf{m}_{2,j}, \mathbf{P}_{2,i}, \mathbf{P}_{2,j}). \quad (2.12)$$

Therefore, by substituting the results of Eqs. (2.10)–(2.12) into Eq. (2.8), we arrive at the final result for the  $L_2$  distance between two GMM pdfs as

$$\begin{aligned} L_2(p_1, p_2) = & \sum_{i=1}^{k_1} \sum_{j=1}^{k_1} w_{1,i} w_{1,j} K(\mathbf{m}_{1,i}, \mathbf{m}_{1,j}, \mathbf{P}_{1,i} + \mathbf{P}_{1,j}) \\ & + \sum_{i=1}^{k_2} \sum_{j=1}^{k_2} w_{2,i} w_{2,j} K(\mathbf{m}_{2,i}, \mathbf{m}_{2,j}, \mathbf{P}_{2,i} + \mathbf{P}_{2,j}) \\ & - 2 \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} w_{1,i} w_{2,j} K(\mathbf{m}_{1,i}, \mathbf{m}_{2,j}, \mathbf{P}_{1,i} + \mathbf{P}_{2,j}) \end{aligned} \quad (2.13)$$

It is seen that for GMMs, the  $L_2$  distance offers a completely closed-form expression of the distance, whereas the same cannot be said for the Kullback-Leibler divergence, in which case approximate values can be obtained by resorting to multi-dimensional numerical integration techniques.<sup>13</sup> This offers a significant advantage of the  $L_2$  approach to computing distance between GMM pdfs as it allows for rapid and accurate calculation of the resultant distance. The  $L_2$  distance reaches its minimum value of zero in the case where the two GMM pdfs are identical and its maximum value when the overlap between the GMM pdfs is zero everywhere.<sup>72</sup>



### 2.2.1.2 Normalized $L_2$ Distance between GMMs

From the triangle inequality property of the  $L_2$  distance, it is known that

$$\int_{\mathbb{R}^n} (p_1(\mathbf{x}) - p_2(\mathbf{x}))^2 d\mathbf{x} \leq \int_{\mathbb{R}^n} p_1^2(\mathbf{x}) d\mathbf{x} + \int_{\mathbb{R}^n} p_2^2(\mathbf{x}) d\mathbf{x},$$

which means that by defining the normalized  $L_2$  distance to be

$$\text{NL}_2(p_1, p_2) = \frac{\int_{\mathbb{R}^n} (p_1(\mathbf{x}) - p_2(\mathbf{x}))^2 d\mathbf{x}}{\int_{\mathbb{R}^n} p_1^2(\mathbf{x}) d\mathbf{x} + \int_{\mathbb{R}^n} p_2^2(\mathbf{x}) d\mathbf{x}}, \quad (2.14)$$

a distance measure which has a minimum value of zero and a maximum value of one is obtained, i.e.  $0 \leq \text{NL}_2(p_1, p_2) \leq 1$ .<sup>52</sup> Expanding the numerator in Eq. (2.14) and rearranging terms, it is seen that the normalized  $L_2$  can be written as

$$\text{NL}_2(p_1, p_2) = 1 - \frac{2 \int_{\mathbb{R}^n} p_1(\mathbf{x}) p_2(\mathbf{x}) d\mathbf{x}}{\int_{\mathbb{R}^n} p_1^2(\mathbf{x}) d\mathbf{x} + \int_{\mathbb{R}^n} p_2^2(\mathbf{x}) d\mathbf{x}}. \quad (2.15)$$

Substituting the results of Eqs. (2.10)–(2.12) into Eq. (2.15), the  $\text{NL}_2$  distance between two GMM pdfs is given by

$$\text{NL}_2(p_1, p_2) = 1 - \frac{2d_{1,2}}{d_{1,1} + d_{2,2}},$$

with

$$\begin{aligned} d_{1,2} &= \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} w_{1,i} w_{2,j} K(\mathbf{m}_{1,i}, \mathbf{m}_{2,j}, \mathbf{P}_{1,i} + \mathbf{P}_{2,j}) \\ d_{1,1} &= \sum_{i=1}^{k_1} \sum_{j=1}^{k_1} w_{1,i} w_{1,j} K(\mathbf{m}_{1,i}, \mathbf{m}_{1,j}, \mathbf{P}_{1,i} + \mathbf{P}_{1,j}) \\ d_{2,2} &= \sum_{i=1}^{k_2} \sum_{j=1}^{k_2} w_{2,i} w_{2,j} K(\mathbf{m}_{2,i}, \mathbf{m}_{2,j}, \mathbf{P}_{2,i} + \mathbf{P}_{2,j}). \end{aligned}$$

As was the case with the  $L_2$  distance, the  $NL_2$  distance offers a completely closed-form expression of the distance between two GMM pdfs. The  $NL_2$  distance provides a scale-invariant form of the  $L_2$  distance which can be viewed as a more intuitive interpretation of the distance between the GMM pdfs. The  $NL_2$  reaches its minimum value of zero in the case where the two GMM pdfs are identical and its maximum value of one when the overlap between the GMM pdfs is zero everywhere.

### 2.2.2 Likelihood Agreement between Distributions

In cases where a sample of data points is to be compared against a representation of the pdf, the  $L_2$  distance or  $NL_2$  distance cannot be used as developed. Even though a set of sample points can be considered as a Dirac mixture model (DMM) representation of the pdf and a DMM is a limiting case of the GMM, the computations of the  $L_2$  and  $NL_2$  distances becomes ill-conditioned. To replace the distance comparison methodology, consider instead a likelihood agreement measure between two pdfs as

$$L(p_1, p_2) = \int_{\mathbb{R}^n} p_1(\mathbf{x})p_2(\mathbf{x})d\mathbf{x} . \quad (2.16)$$

The likelihood measure,  $L$ , describes the amount of overlap between the two densities and will therefore be larger for densities that are in greater agreement with one another. Since we are, in this case, interested in the agreement between a set of sampled data points and a probability density function, we let  $p_1(\mathbf{x})$  be given by the DMM

$$p_1(\mathbf{x}) = \sum_{i=1}^{k_1} w_{1,i} \delta(\mathbf{x} - \mathbf{m}_{1,i}) , \quad (2.17)$$

where  $\delta(\mathbf{x} - \mathbf{m}_{1,i})$  is a Dirac delta distribution centered at  $\mathbf{m}_{1,i}$  with weight  $w_{1,i}$ , and the Dirac delta is defined such that it is zero everywhere except at  $\mathbf{m}_{1,i}$ . Furthermore, the Dirac delta satisfies the integral condition

$$\int_{\mathbb{R}^n} \delta(\mathbf{x} - \mathbf{m}_{1,i}) d\mathbf{x} = 1. \quad (2.18)$$

From the integral property of the Dirac delta in Eq. (2.18) and the definition of the DMM in Eq. (2.17) it is readily observed that the weights must satisfy  $\sum_{i=1}^{k_1} w_{1,i} = 1$ , where usually the sample points are equally weighted, that is  $w_{1,i} = 1/k_1 \quad \forall \quad i \in \{1, \dots, k_1\}$ . Substituting the DMM of Eq. (2.17) into the likelihood agreement measure of Eq. (2.16) yields

$$L(p_1, p_2) = \sum_{i=1}^{k_1} w_{1,i} \int_{\mathbb{R}^n} p_2(\mathbf{x}) \delta(\mathbf{x} - \mathbf{m}_{1,i}) d\mathbf{x}.$$

Applying the sifting property of the Dirac delta,<sup>34</sup> it is seen that the likelihood agreement measure can be written as

$$L(p_1, p_2) = \sum_{i=1}^{k_1} w_{1,i} p_2(\mathbf{m}_{1,i}),$$

for any arbitrary comparison pdf  $p_2(\mathbf{x})$ . If the comparison pdf is given by a GMM of the form

$$p_2(\mathbf{x}) = \sum_{j=1}^{k_2} w_{2,j} p_g(\mathbf{x}; \mathbf{m}_{2,j}, \mathbf{P}_{2,j}),$$

then the likelihood agreement measure can be rewritten explicitly in terms of the individual weights, means, and covariances of the GMM as

$$L(p_1, p_2) = \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} w_{1,i} w_{2,j} p_g(\mathbf{m}_{1,i}; \mathbf{m}_{2,j}, \mathbf{P}_{2,j}). \quad (2.19)$$

Therefore, given a set of sample points via a DMM and a GMM upon which to compare the samples to, the likelihood that the DMM represents the same distribution as the GMM can be computed via Eq. (2.19). A higher value of the likelihood agreement will indicate that a given GMM was more likely to have generated the DMM, thereby allowing multiple GMMs to be compared for accuracy to a single DMM, with the most accurate GMM having the highest value of the likelihood agreement.

### 2.2.3 Differential Entropy and Rényi Entropy of a Gaussian Distribution

One of the final pdf measure to be discussed is the differential entropy, which is a measure of the average amount of surprisal in a random variable. Given any pdf  $p(\mathbf{x})$ , the differential entropy is defined by<sup>10,30,57</sup>

$$H(\mathbf{x}) = - \int_{\mathbb{R}^n} p(\mathbf{x}) \log p(\mathbf{x}) d\mathbf{x} ,$$

or, alternatively, this can be expressed in terms of the expected value with respect to  $p(\mathbf{x})$  of the negative logarithm of  $p(\mathbf{x})$ , yielding

$$H(\mathbf{x}) = E \{ - \log p(\mathbf{x}) \} .$$

Having the definition of the differential entropy in hand, we now develop an equation for evaluating this quantity for a Gaussian pdf, which has the form

$$p(\mathbf{x}) = |2\pi \mathbf{P}|^{-1/2} \exp \left\{ -\frac{1}{2}(\mathbf{x} - \mathbf{m})^T \mathbf{P}^{-1}(\mathbf{x} - \mathbf{m}) \right\} . \quad (2.20)$$

Therefore, by taking the negative logarithm (base  $e$ ) of Eq. (2.20), it follows that

$$\begin{aligned}
-\log p(\mathbf{x}) &= \frac{1}{2} \log |2\pi \mathbf{P}| + \frac{1}{2} (\mathbf{x} - \mathbf{m})^T \mathbf{P}^{-1} (\mathbf{x} - \mathbf{m}) \\
&= \frac{1}{2} \log |2\pi \mathbf{P}| + \frac{1}{2} \text{trace} \{ (\mathbf{x} - \mathbf{m})^T \mathbf{P}^{-1} (\mathbf{x} - \mathbf{m}) \} \\
&= \frac{1}{2} \log |2\pi \mathbf{P}| + \frac{1}{2} \text{trace} \{ (\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T \mathbf{P}^{-1} \} , \tag{2.21}
\end{aligned}$$

where we have used the invariance under cyclic permutation property of the trace operator.<sup>3</sup> By then taking the expected value of Eq. (2.21) we find that

$$\begin{aligned}
\mathbb{E} \{ -\log p(\mathbf{x}) \} &= \frac{1}{2} \log |2\pi \mathbf{P}| + \frac{1}{2} \text{trace} \{ \mathbf{P} \mathbf{P}^{-1} \} \\
&= \frac{1}{2} \log |2\pi \mathbf{P}| + \frac{1}{2} n .
\end{aligned}$$

Noting that  $\frac{1}{2}n = \frac{1}{2}n \log e$  and rearranging terms, the final form of the differential entropy for a Gaussian distribution is given in terms of the logarithm of the determinant of a scaled form of the covariance matrix, that is

$$H(\mathbf{x}) = \frac{1}{2} \log |2\pi e \mathbf{P}| \tag{2.22}$$

Alternatively, by noting that the determinant of the covariance matrix is given by the product of the eigenvalues, Eq. (2.22) can be equivalently expressed as

$$H(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^n \log(2\pi e \lambda_i) ,$$

where  $n$  is the dimension of the random variable  $\mathbf{x}$  and  $\lambda_i$  is the  $i^{\text{th}}$  eigenvalue of  $\mathbf{P}$ .

A generalization of the differential entropy is that of the Rényi entropy, which allows for different averaging of probabilities through a control parameter  $\kappa$ . The

Rényi entropy of order  $\kappa$  for a continuous random variable with pdf  $p(\mathbf{x})$  is defined by<sup>53, 60, 77</sup>

$$R_\kappa(\mathbf{x}) = \frac{1}{1-\kappa} \log \int_{\mathbb{R}^n} p(\mathbf{x})^\kappa d\mathbf{x}, \quad (2.23)$$

for  $\kappa > 0$ ,  $\kappa \neq 1$ , and

$$\lim_{\kappa \rightarrow 1} R_\kappa(\mathbf{x}) = H(\mathbf{x}). \quad (2.24)$$

Now, we consider the form of the Rényi entropy for the case of a Gaussian pdf, which is defined as

$$p(\mathbf{x}) = |2\pi\mathbf{P}|^{-1/2} \exp \left\{ -\frac{1}{2}(\mathbf{x} - \mathbf{m})^T \mathbf{P}^{-1}(\mathbf{x} - \mathbf{m}) \right\}. \quad (2.25)$$

From Eq. (2.25), it is then straightforward to see that  $p(\mathbf{x})^\kappa$  is

$$p(\mathbf{x})^\kappa = |2\pi\mathbf{P}|^{-\kappa/2} \exp \left\{ -\frac{\kappa}{2}(\mathbf{x} - \mathbf{m})^T \mathbf{P}^{-1}(\mathbf{x} - \mathbf{m}) \right\}. \quad (2.26)$$

Given that  $p(\mathbf{x})$  is a pdf and therefore when integrated across the support of the pdf must be one, it is observed that

$$\int_{\mathbb{R}^n} \exp \left\{ -\frac{\kappa}{2}(\mathbf{x} - \mathbf{m})^T \mathbf{P}^{-1}(\mathbf{x} - \mathbf{m}) \right\} d\mathbf{x} = \kappa^{-n/2} |2\pi\mathbf{P}|^{1/2}, \quad (2.27)$$

where  $n$  is the dimension of the random variable  $\mathbf{x}$ . Therefore, by Eqs. (2.26) and (2.27), we have that

$$\int_{\mathbb{R}^n} p(\mathbf{x})^\kappa d\mathbf{x} = \kappa^{-n/2} |2\pi\mathbf{P}|^{-\kappa/2} |2\pi\mathbf{P}|^{1/2},$$

which allows the Rényi entropy to be found for a Gaussian pdf from Eq. (2.23) to be

$$R_\kappa(\mathbf{x}) = \frac{1}{2} \log |2\pi\mathbf{P}| - \frac{n}{2(1-\kappa)} \log \kappa. \quad (2.28)$$

After some manipulation of Eq. (2.28), a more convenient form of the Rényi entropy is given by the logarithm of the determinant of a scaled form of the covariance matrix, that is

$$R_\kappa(\mathbf{x}) = \frac{1}{2} \log \left| 2\pi\kappa^{\frac{1}{\kappa-1}} \mathbf{P} \right|. \quad (2.29)$$

In the same manner as the differential entropy, an alternative representation can be found by noting that the determinant of the covariance matrix is given by the product of the eigenvalues, such that Eq. (2.29) can be equivalently expressed as

$$R_\kappa(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^n \log(2\pi\kappa^{\frac{1}{\kappa-1}} \lambda_i),$$

where  $n$  is the dimension of the random variable  $\mathbf{x}$  and  $\lambda_i$  is the  $i^{\text{th}}$  eigenvalue of  $\mathbf{P}$ .

It is worth noting that

$$\lim_{\kappa \rightarrow 1} \kappa^{\frac{1}{\kappa-1}} = e,$$

such that the claim in Eq. (2.24) is verified for a Gaussian pdf by a simple comparison of the results obtained on differential entropy with those on Rényi entropy.

## 2.3 Splitting a Gaussian Distribution

Splitting a Gaussian distribution into “smaller” distributions is a subject that has received more attention as GMMs have become more widely used. For instance, Hanebeck illustrates a method for splitting a univariate Gaussian into two components<sup>17</sup> and a method for splitting a univariate Gaussian in multiple components.<sup>16</sup> Li, on the other hand gives a method for splitting a multivariate Gaussian into two

components.<sup>36,37</sup> Finally, Huber, building upon the work of Hanebeck, utilizes pre-computed libraries of univariate multi-component splitting to develop a multivariate Gaussian splitting technique.<sup>18</sup> This technique enables reliable, accurate splitting of multivariate Gaussian distributions into many components. Therefore, following in the spirit of Huber’s method, univariate Gaussian splitting libraries are developed and then applied to the multivariate Gaussian case. However, the approach presented here for developing the univariate Gaussian splitting libraries is different than that of Huber’s, and the application to the multivariate Gaussian case has been generalized from previous approaches.

### 2.3.1 The Univariate Case

As a precursor to developing a method for splitting a multivariate Gaussian distribution, consider first the splitting of a univariate Gaussian distribution. Without loss of generality, since all univariate Gaussian distributions can be brought to the so-called “standard” Gaussian distribution (that is a Gaussian distribution with zero mean and unit variance) by a linear transformation, it is desired to approximate the standard Gaussian distribution,  $p_1(x)$ , with a GMM distribution,  $p_2(x)$ . That is, we want to approximate

$$p_1(x) = p_g(x; 0, 1) \tag{2.30}$$

by a GMM distribution of the form

$$p_2(x) = \sum_{i=1}^{k_2} \tilde{w}_i p_g(x; \tilde{m}_i, \tilde{\sigma}^2), \tag{2.31}$$



where it should be noted that the variances of the GMM components have been constrained to be equal. If each component is allowed to have a different variance, then situations arise in the splitting process where some components may take on a very large variance and other components may take on a very small variance. The goal therefore is to create a GMM approximation to the standard Gaussian distribution where the “width” of all of the components is equal. In order to find the parameters of the GMM distribution, we view this as a minimization problem where we wish to minimize the distance between  $p_1(x)$  and  $p_2(x)$ . Additionally, it is desired that the single variance parameter  $\tilde{\sigma}^2$  is small, such that combining this condition with the minimum distance condition, a performance index can be stated as

$$J = L_2(p_1, p_2) + \lambda \tilde{\sigma}^2 \quad \text{subject to} \quad \sum_{i=1}^{k_2} \tilde{w}_i = 1 \quad (2.32)$$

where  $\lambda$  is a weighting term that scales the importance of minimizing  $\tilde{\sigma}^2$  versus minimizing  $L_2(p_1, p_2)$ . Finally, in Eq. (2.32), we have also noted the constraint imposed by the GMM distribution weights; that is, the sum of the weights must be one in order for  $p_2(x)$  to be a valid pdf.

With the above method for splitting a standard Gaussian distribution we compute the splitting libraries for  $k_2 = 3$ ,  $k_2 = 4$ , and  $k_2 = 5$ . The choice of the scaling term  $\lambda$ , the achieved  $L_2$  distance between  $p_1(x)$  and  $p_2(x)$ , and the value of  $\sigma_2$  are summarized for each of the computed splitting libraries in Table 2.1. While the method proposed is general for any value of  $k_2$ , it is seen that as  $k_2$  grows larger, there is a diminishing return in reducing the  $\tilde{\sigma}$  value using higher-component splits as can be seen in Table 2.1. For this reason, only splitting libraries up to  $k_2 = 5$  are presented.

Table 2.1: Parameters and Performance of Splitting Libraries

$k_2$	$\lambda$	$L_2(p_1, p_2)$	$\tilde{\sigma}$
3	0.0010	$6.139 \times 10^{-5}$	0.6715662886640760
4	0.0020	$9.566 \times 10^{-5}$	0.5276007226175397
5	0.0025	$5.216 \times 10^{-5}$	0.4422555386310084

The computed values of the component weights, means, and standard deviations is given for  $k_2 = 3$  in Table 2.2, for  $k_2 = 4$  in Table 2.3, and for  $k_2 = 5$  in Table 2.4. Finally, to graphically illustrate the efficacy of the method, the original target distribution ( $p_1(x)$ ), the individual computed components of the split distribution, and the overall split distribution ( $p_2(x)$ ) are shown in Figures 2.1–2.3 for the three values of  $k_2$  considered.

Table 2.2: 3-Component Splitting Library

$i$	$\tilde{w}_i$	$\tilde{m}_i$	$\tilde{\sigma}$
1	0.2252246249136750	$-1.057515461475881$	0.6715662886640760
2	0.5495507501726501	0	0.6715662886640760
3	0.2252246249136750	$1.057515461475881$	0.6715662886640760

Table 2.3: 4-Component Splitting Library

$i$	$\tilde{w}_i$	$\tilde{m}_i$	$\tilde{\sigma}$
1	0.1238046161618835	-1.437464136328835	0.5276007226175397
2	0.3761953838381165	-0.455886223973523	0.5276007226175397
3	0.3761953838381165	0.455886223973523	0.5276007226175397
4	0.1238046161618835	1.437464136328835	0.5276007226175397

Table 2.4: 5-Component Splitting Library

$i$	$\tilde{w}_i$	$\tilde{m}_i$	$\tilde{\sigma}$
1	0.0763216490701042	-1.689972911128078	0.4422555386310084
2	0.2474417859474436	-0.800928383429953	0.4422555386310084
3	0.3524731299649044	0	0.4422555386310084
4	0.2474417859474436	0.800928383429953	0.4422555386310084
5	0.0763216490701042	1.689972911128078	0.4422555386310084

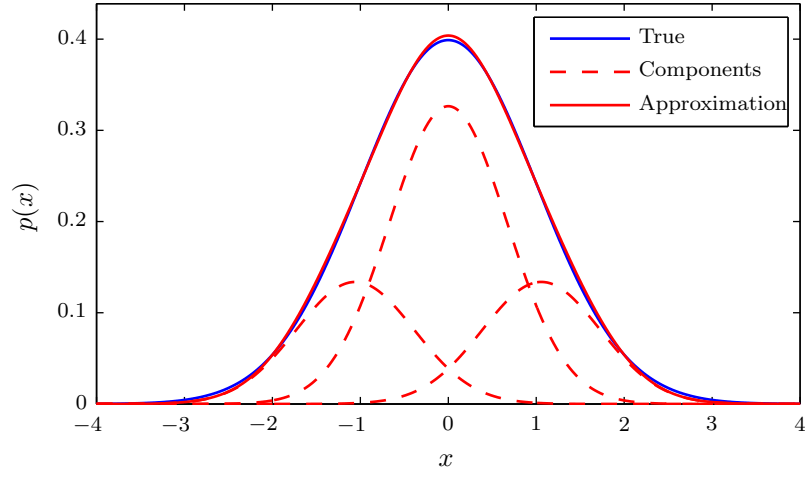


Figure 2.1: Components of the splitting libraries and their sum as compared to the normal Gaussian distribution for  $k_2 = 3$ .

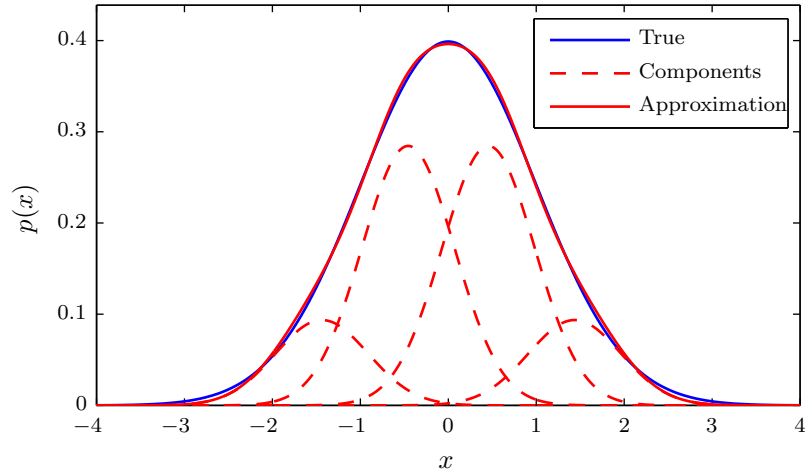


Figure 2.2: Components of the splitting libraries and their sum as compared to the normal Gaussian distribution for  $k_2 = 4$ .

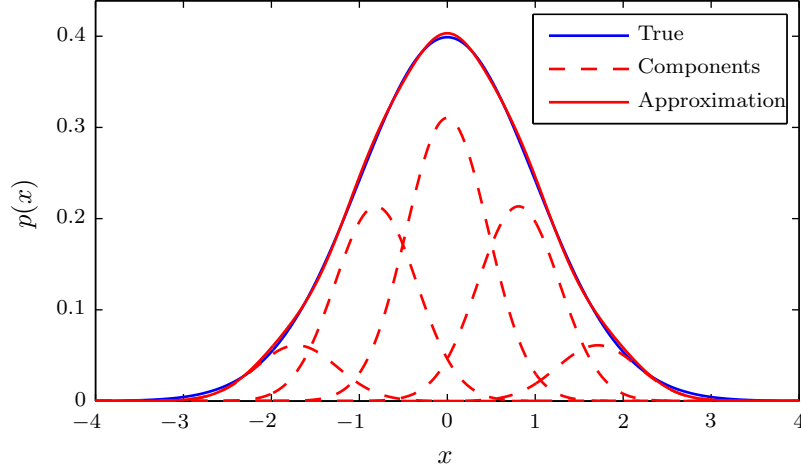


Figure 2.3: Components of the splitting libraries and their sum as compared to the normal Gaussian distribution for  $k_2 = 5$ .

### 2.3.2 The Multivariate Case

Consider the case where it is desired to replace a component of a GMM using a splitting process. In this case, the goal is to find the  $N$  component weights, means, and covariances which, when combined in a GMM, yield the same approximate pdf as the original component, that is

$$wp_g(\mathbf{x}; \mathbf{m}, \mathbf{P}) \approx \sum_{i=1}^N w_i p_g(\mathbf{x}; \mathbf{m}_i, \mathbf{P}_i). \quad (2.33)$$

To apply a univariate splitting library to the multivariate case, the approximation must be applied in a specified direction. The best way to think of this is to consider the principal directions of the covariance matrix (given by the eigenvectors of the covariance matrix). Then, in the coordinate system described by the principal directions, the multivariate Gaussian distribution becomes a product of univariate Gaus-

sian distributions, which allows for the easy implementation of a univariate splitting technique to be applied to any one, several, or all of the elements in this product of univariate Gaussian distributions. While thinking of the principal directions provides nice physical insight into the problem, it is not required for describing the general approach.

To apply the univariate Gaussian splitting technique, first find a square-root factor  $\mathbf{S}$  such that  $\mathbf{S}\mathbf{S}^T = \mathbf{P}$ . Then, separate the square-root factor into its columns, such that  $\mathbf{s}_k$  is the  $k^{\text{th}}$  column of  $\mathbf{S}$ . Select the square-root factor column upon which the univariate splitting is to be performed, as well as the splitting library to be used (e.g. a 3-component, 4-component, or 5-component library) which specifies values for  $\tilde{w}_i$ ,  $\tilde{m}_i$ , and  $\tilde{\sigma}$ . Then, when the splitting is performed along the  $k^{\text{th}}$  axis of the square-root factor, the component weights, means, and covariances to be used in Eq. (2.33) are given by

$$w_i = \tilde{w}_i w$$

$$\mathbf{m}_i = \mathbf{m} + \tilde{m}_i \mathbf{s}_k$$

$$\mathbf{P}_i = \mathbf{S}_i \mathbf{S}_i^T,$$

where  $\mathbf{S}_i$  is the square-root factor of the  $i^{\text{th}}$  new component, which is

$$\mathbf{S}_i = [\mathbf{s}_1, \dots, \tilde{\sigma} \mathbf{s}_k, \dots, \mathbf{s}_n].$$

One downside of the general approach using an arbitrary square-root factor is that the physical meaning of the directions along which the univariate splitting is applied is lost. However, if we choose the specific square-root factor to be one

formed by the eigenvalues and eigenvectors, then physical meaning is re-established. Therefore, consider the spectral factorization of the covariance matrix  $\mathbf{P}$  as

$$\mathbf{P} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T ,$$

for which the square-root factor can be readily determined as  $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}^{1/2}$ . Since  $\mathbf{\Lambda}$  is diagonal,  $\mathbf{\Lambda}^{1/2}$  is well defined. Using the spectral factorization, an eigenvector (along which the splitting is to be done) is selected and the splitting library to be used is selected. Applying the square-root factor from the spectral factorization to the general case, it is seen that when the splitting is performed along the  $k^{\text{th}}$  axis of the spectral factorization, the component weights, means, and covariances to be used in Eq. (2.33) become

$$\begin{aligned} w_i &= \tilde{w}_i w \\ \mathbf{m}_i &= \mathbf{m} + \sqrt{\lambda_k} \tilde{m}_i \mathbf{v}_k \\ \mathbf{P}_i &= \mathbf{V}\mathbf{\Lambda}_i\mathbf{V}^T , \end{aligned}$$

where  $\mathbf{v}_k$  is the  $k^{\text{th}}$  eigenvector of  $\mathbf{P}$  and  $\mathbf{\Lambda}_i$  is the set of eigenvalues of the  $i^{\text{th}}$  new component, given by

$$\mathbf{\Lambda}_i = \text{diag} \{ \lambda_1 , \dots , \tilde{\sigma}^2 \lambda_k , \dots , \lambda_n \} .$$

Using the spectral factorization to generate the square-root factor leads to an algorithm equivalent to that of Huber.<sup>18</sup>

## 2.4 Merging Gaussian Distributions

With a GMM, the situation arises where multiple components become redundant and can be alternatively well-represented by a single merged component. The most common method of merging this set of redundant components is to compute a distance measure between pairs of components, such as the Kullback-Leibler divergence, the  $L_2$  distance, or the  $NL_2$  distance, and combine the two components with the smallest distance into a single merged-component. This process is then repeated on the new set of components until there are no more possible mergers (either because no components are close enough together, or because there is only one component left).<sup>72,74</sup> The procedure of iteratively combined components pair-wise is known here as the “bottom-up method” since it works by considering all possible merges of two components. An alternative method, which is referred to here as the “top-down method,” approaches the problem from the consideration that perhaps there is a set of more than two components which can be merged together. In this approach a subset of components is proposed for possible merging and the distance between the reduced GMM and the original GMM is computed and stored. Once all subsets of equal subset size are considered, the proposed merge with the lowest distance is taken as the accepted merge. The process is then repeated on the new GMM until no acceptable merging remains.

### 2.4.1 Method of Moments

Before describing the top-down method for merging components, we first consider how to merge multiple components. To accomplish this multiple component



merging, the method of moments is detailed.<sup>63</sup> Consider a GMM of the form

$$p(\mathbf{x}) = \sum_{k=1}^K w_k p_g(\mathbf{x}; \mathbf{m}_k, \mathbf{P}_k), \quad (2.34)$$

where the component weights, means, and covariances are given, respectively, by  $w_k$ ,  $\mathbf{m}_k$ , and  $\mathbf{P}_k$ . The goal of merging is to replace the GMM of Eq. (2.34) with a single component with weight  $w_m$ , mean  $\mathbf{m}_m$ , and covariance  $\mathbf{P}_m$ , that is

$$\sum_{k=1}^K w_k p_g(\mathbf{x}; \mathbf{m}_k, \mathbf{P}_k) = w_m p_g(\mathbf{x}; \mathbf{m}_m, \mathbf{P}_m). \quad (2.35)$$

The overall weight must remain the same, and therefore the merged weight  $w_m$  is given by the sum of the component weights, i.e.

$$w_m = \sum_{k=1}^K w_k. \quad (2.36)$$

Computing the expected value of both sides of Eq. (2.35) in order to determine the merged weight yields

$$\int_{\mathbb{R}^n} \mathbf{x} \sum_{k=1}^K w_k p_g(\mathbf{x}; \mathbf{m}_k, \mathbf{P}_k) d\mathbf{x} = \int_{\mathbb{R}^n} \mathbf{x} w_m p_g(\mathbf{x}; \mathbf{m}_m, \mathbf{P}_m) d\mathbf{x},$$

or

$$\sum_{k=1}^n w_k \int_{\mathbb{R}^n} \mathbf{x} p_g(\mathbf{x}; \mathbf{m}_k, \mathbf{P}_k) d\mathbf{x} = w_m \int_{\mathbb{R}^n} \mathbf{x} p_g(\mathbf{x}; \mathbf{m}_m, \mathbf{P}_m) d\mathbf{x}. \quad (2.37)$$

Making use of the properties of the Gaussian distribution, Eq. (2.37) reduces to

$$\sum_{k=1}^n w_k \mathbf{m}_k = w_m \mathbf{m}_m,$$

which is readily solved for the merged mean yielding

$$\mathbf{m}_m = \frac{1}{w_m} \sum_{k=1}^K w_k \mathbf{m}_k. \quad (2.38)$$

In a similar manner, the method of moments considers the raw second moment of both sides of Eq. (2.35) and equates them in to determine the merged covariance.

Computing the raw second moment of both sides of Eq. (2.35) gives

$$\int_{\mathbb{R}^n} \mathbf{x} \mathbf{x}^T \sum_{k=1}^K w_k p_g(\mathbf{x}; \mathbf{m}_k, \mathbf{P}_k) d\mathbf{x} = \int_{\mathbb{R}^n} \mathbf{x} \mathbf{x}^T w_m p_g(\mathbf{x}; \mathbf{m}_m, \mathbf{P}_m) d\mathbf{x},$$

or

$$\sum_{k=1}^K w_k \int_{\mathbb{R}^n} \mathbf{x} \mathbf{x}^T p_g(\mathbf{x}; \mathbf{m}_k, \mathbf{P}_k) d\mathbf{x} = w_m \int_{\mathbb{R}^n} \mathbf{x} \mathbf{x}^T p_g(\mathbf{x}; \mathbf{m}_m, \mathbf{P}_m) d\mathbf{x}. \quad (2.39)$$

Again, applying the properties of the Gaussian distribution, Eq. (2.39) becomes

$$\sum_{k=1}^K w_k (\mathbf{P}_k + \mathbf{m}_k \mathbf{m}_k^T) = w_m (\mathbf{P}_m + \mathbf{m}_m \mathbf{m}_m^T),$$

which is readily solved for the merged covariance, as

$$\mathbf{P}_m = \frac{1}{w_m} \sum_{k=1}^K w_k (\mathbf{P}_k + \mathbf{m}_k \mathbf{m}_k^T) - \mathbf{m}_m \mathbf{m}_m^T. \quad (2.40)$$

Collecting the results of Eqs. (2.36) , (2.38) , and (2.40), the merged weight, mean, and covariance of a GMM as determined by the method of moments are given by

$$\begin{aligned} w_m &= \sum_{k=1}^n w_k \\ \mathbf{m}_m &= \sum_{k=1}^n \frac{w_k}{w_m} \mathbf{m}_k \\ \mathbf{P}_m &= \sum_{k=1}^n \frac{w_k}{w_m} (\mathbf{P}_k + \mathbf{m}_k \mathbf{m}_k^T) - \mathbf{m}_m \mathbf{m}_m^T. \end{aligned}$$

### 2.4.2 Top-Down Merging Method

Consider a  $K_1$ -component GMM distribution described by

$$p_1(\mathbf{x}) = \sum_{i=1}^{K_1} w_i p_g(\mathbf{x}; \mathbf{m}_i, \mathbf{P}_i).$$

Merging of components seeks to determine a  $K_2$ -component GMM distribution of the form

$$p_2(\mathbf{x}) = \sum_{j=1}^{K_2} w_j p_g(\mathbf{x}; \mathbf{m}_j, \mathbf{P}_j),$$

with  $K_2 < K_1$ , such that the distance between  $p_1(\mathbf{x})$  and  $p_2(\mathbf{x})$  is small. Let  $\mathcal{I}_1 = \{1, 2, \dots, K_1\}$ , such that  $p_1(\mathbf{x})$  can be alternatively expressed as

$$p_1(\mathbf{x}) = \sum_{i \in \mathcal{I}_1} w_i p_g(\mathbf{x}; \mathbf{m}_i, \mathbf{P}_i).$$

Furthermore, if  $\mathcal{I}_2 \subseteq \mathcal{I}_1$ , then  $p_1(\mathbf{x})$  can also be written as

$$p_1(\mathbf{x}) = \sum_{i \in \mathcal{I}_2} w_i p_g(\mathbf{x}; \mathbf{m}_i, \mathbf{P}_i) + \sum_{i \in \mathcal{I}_1 \setminus \mathcal{I}_2} w_i p_g(\mathbf{x}; \mathbf{m}_i, \mathbf{P}_i), \quad (2.41)$$

where  $\mathcal{I}_1 \setminus \mathcal{I}_2$  represents the subtraction of  $\mathcal{I}_2$  from  $\mathcal{I}_1$ . It is the first term in Eq. (2.41) that is now considered for merging, implying that  $p_2(\mathbf{x})$  is given by

$$p_2(\mathbf{x}) = w_m p_g(\mathbf{x}; \mathbf{m}_m, \mathbf{P}_m) + \sum_{i \in \mathcal{I}_1 \setminus \mathcal{I}_2} w_i p_g(\mathbf{x}; \mathbf{m}_i, \mathbf{P}_i),$$

where

$$\begin{aligned} w_m &= \sum_{i \in \mathcal{I}_2} w_i \\ \mathbf{m}_m &= \sum_{i \in \mathcal{I}_2} \frac{w_i}{w_m} \mathbf{m}_i \\ \mathbf{P}_m &= \sum_{i \in \mathcal{I}_2} \frac{w_i}{w_m} (\mathbf{P}_i + \mathbf{m}_i \mathbf{m}_i^T) - \mathbf{m}_m \mathbf{m}_m^T. \end{aligned}$$

The top-down merging method begins with  $\mathcal{I}_2 = \mathcal{I}_1$ , meaning that the two sets also have the same Cardinality, i.e.  $\#\mathcal{I}_2 = \#\mathcal{I}_1$ , computes the merged weights, means, and covariances of the components in  $\mathcal{I}_2$ , and then computes the  $\text{NL}_2$  distance between  $p_1(\mathbf{x})$  and  $p_2(\mathbf{x})$ . If this distance is less than a specified tolerance, then the merge is accepted and the process is complete since  $p_2(\mathbf{x})$  will have only one component. If not, then the process continues by considering all possible sets  $\mathcal{I}_2$  such that  $\#\mathcal{I}_2 = \#\mathcal{I}_1 - 1$ , which will give

$$c = \frac{\#\mathcal{I}_1!}{\#\mathcal{I}_2!(\#\mathcal{I}_1 - \#\mathcal{I}_2)!}$$

possible sets of components for merging. The merged weights, means, and covariances resulting from the components in each possible  $\mathcal{I}_2$  are determined, followed by the computation of the  $\text{NL}_2$  distance between  $p_1(\mathbf{x})$  and  $p_2(\mathbf{x})$ . If any of the distances fall under the specified tolerance, the minimum distance merger from the  $c$  candidates is accepted and the process terminates. If none of the distances fall under the tolerance, then the process continues by considering all possible sets  $\mathcal{I}_2$  such that  $\#\mathcal{I}_2 = \#\mathcal{I}_1 - 2$  in the same manner. This repeats until the Cardinality of  $\mathcal{I}_2$  becomes 2. If desired, the entire process may be repeated to find any other possible mergers by beginning again using  $p_2(\mathbf{x})$  as the starting GMM distribution.

## Chapter 3

### Recursive Filtering Strategies

Many systems of interest fall under the broad classification of nonlinear systems. An estimation algorithm which exploits at least some characteristics of the nonlinearities is preferable to retracting the problem to that of a linear one. Consider the nonlinear dynamical system governed by the differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad (3.1)$$

where  $\mathbf{x}(t)$  is the state of the system,  $\mathbf{f}(\cdot)$  represents the sufficiently differentiable nonlinear dynamics of the system, and  $\mathbf{x}_0$  is the initial condition. The initial condition is assumed to be random with pdf  $p(\mathbf{x}_0)$ . The lack of a process noise term in Eq. (3.1) is justified on the basis of the nature of the problems that we will consider in Chapter 5, wherein any noise that may be present is insignificant in comparison to the natural dynamics of the problem. The state of the system is indirectly observed by discrete-time nonlinear imperfect measurements at time  $t_k$ , which are described by

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, t_k) + \mathbf{v}_k, \quad (3.2)$$

where  $\mathbf{x}_k$  is the state of the system at time  $t_k$ ,  $\mathbf{h}(\cdot)$  represents the sufficiently differentiable nonlinear measurement function, and  $\mathbf{v}_k$  is the measurement noise. The

measurement noise is assumed to be a zero-mean white-noise sequence with first and second moments given, respectively, by

$$\mathbf{E} \{ \mathbf{v}_k \} = \mathbf{0} \quad \text{and} \quad \mathbf{E} \{ \mathbf{v}_k \mathbf{v}_{k'}^T \} = \mathbf{R}_k \delta_{kk'} ,$$

where  $\delta_{kk'} = 0$  for  $k \neq k'$  and  $\delta_{kk'} = 1$  for  $k = k'$ .

## 3.1 The Extended Kalman Filter

### 3.1.1 Propagation

The propagation stage of the extended Kalman filter (EKF) implements differential equations for the time-wise evolution of the mean and covariance by utilizing a first-order Taylor series expansion of the nonlinear dynamical system about a current best estimate. The EKF implicitly assumes that the deviation of the true system from the best estimate is a zero-mean process which can be well-represented by the covariance centered on the best estimate. To arrive at the propagation equations for the EKF, consider the expected value of the nonlinear dynamical system of Eq. (3.1); that is

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{E} \{ \dot{\mathbf{x}}(t) \} = \mathbf{E} \{ \mathbf{f}(\mathbf{x}(t), t) \} .$$

Expanding the nonlinear dynamics in a first-order Taylor series about an estimate  $\hat{\mathbf{x}}(t)$ , and defining the error between the truth and the estimate to be  $\mathbf{e}(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t)$ , it follows that

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{f}(\hat{\mathbf{x}}(t), t) + \mathbf{E} \{ \mathbf{F}(\hat{\mathbf{x}}(t), t) \mathbf{e}(t) \} ,$$

where the dynamics Jacobian,  $\mathbf{F}(\hat{\mathbf{x}}(t), t)$ , is defined as

$$\mathbf{F}(\hat{\mathbf{x}}(t), t) = \left[ \frac{\partial \mathbf{f}(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)} \right]_{\mathbf{x}(t)=\hat{\mathbf{x}}(t)}.$$

Assuming the estimate to be unbiased (equivalently, assuming that  $\mathbf{e}(t)$  is a zero-mean process) yields the state estimate propagation equation for the EKF:

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{f}(\hat{\mathbf{x}}(t), t), \quad \hat{\mathbf{x}}(t_{k-1}) = \hat{\mathbf{x}}_{k-1}, \quad t_{k-1} \leq t \leq t_k. \quad (3.3)$$

From the given definition of the error between the true state and the state estimate, the linearized dynamics of the error are given by

$$\dot{\mathbf{e}}(t) = \dot{\mathbf{x}}(t) - \dot{\hat{\mathbf{x}}}(t) = \mathbf{F}(\hat{\mathbf{x}}(t), t)\mathbf{e}(t),$$

which has the solution<sup>55</sup>

$$\mathbf{e}(t) = \Phi(t, t_{k-1})\mathbf{e}(t_{k-1}), \quad (3.4)$$

where  $\Phi(t, t_{k-1})$  is the state transition matrix satisfying

$$\dot{\Phi}(t, t_{k-1}) = \mathbf{F}(\hat{\mathbf{x}}(t), t)\Phi(t, t_{k-1}), \quad \Phi(t_{k-1}, t_{k-1}) = \mathbf{I}.$$

Since  $\mathbf{e}(t)$  is a zero-mean process, the state estimation error covariance is found via

$$\mathbf{P}(t) = \mathbb{E} \{ \mathbf{e}(t)\mathbf{e}^T(t) \},$$

such that, by substitution of Eq. (3.4), we have

$$\mathbf{P}(t) = \Phi(t, t_{k-1})\mathbf{P}(t_{k-1})\Phi^T(t, t_{k-1}), \quad (3.5)$$

which guarantees that for positive definite, symmetric  $\mathbf{P}(t_{k-1})$ ,  $\mathbf{P}(t)$  remains symmetric and positive definite.<sup>35</sup> Given a state estimate and covariance at time  $t_{k-1}$  as

$\hat{\mathbf{x}}_{k-1}$  and  $\mathbf{P}_{k-1}$ , respectively, the propagation of the state estimate is accomplished by numerically integrating Eq. (3.3), while the propagation of the covariance is accomplished by numerically integrating the state transition matrix with Eq. (3.4), and then mapping the covariance forward using Eq. (3.5).

### 3.1.2 Update

At time  $t_k$ , information is made available via the measurement  $\mathbf{y}_k$ . This information is used in conjunction with the prior information regarding the state, i.e. the *a priori* state estimate  $\hat{\mathbf{x}}_k^-$  and state estimation error covariance  $\mathbf{P}_k^-$ , to yield a blending of the new information with the existing information. To this extent, assume a linear update which blends the *a priori* state estimate with the measurement via

$$\hat{\mathbf{x}}_k^+ = \mathbf{L}_k \hat{\mathbf{x}}_k^- + \mathbf{K}_k \mathbf{y}_k. \quad (3.6)$$

Let the *a priori* (denoted by superscript “−”) and *a posteriori* (denoted by superscript “+”) estimation errors be defined as

$$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^- \quad \text{and} \quad \mathbf{e}_k^+ = \mathbf{x}_k - \hat{\mathbf{x}}_k^+.$$

By the definition of  $\mathbf{e}_k^-$  and  $\mathbf{e}_k^+$ , the *a posteriori* state estimate is given by

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{e}_k^- - \mathbf{e}_k^+. \quad (3.7)$$

Substituting Eq. (3.7) into Eq. (3.6), the linear update equation can be written as

$$-\mathbf{e}_k^+ = \mathbf{L}_k \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{h}(\mathbf{x}(t_k), t_k) + \mathbf{v}_k) - \mathbf{e}_k^- + \hat{\mathbf{x}}_k^-$$



By taking the expected value of both sides, noting that the *a priori* state estimate is unbiased (based on the previous discussion regarding the propagation phase of the EKF) and forcing the *a posteriori* estimation error to be unbiased, it is found that

$$\mathbf{L}_k \hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}_k^- - \mathbf{K}_k \mathbb{E} \{ \mathbf{h}(\mathbf{x}(t_k), t_k) \} ,$$

which can then be substituted into Eq. (3.6) yielding

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbb{E} \{ \mathbf{h}(\mathbf{x}(t_k), t_k) \}) .$$

Expanding the nonlinear measurement function,  $\mathbf{h}(\cdot)$ , in a first-order Taylor series about the *a priori* state estimate and using the unbiased nature of the state estimate, the linear update equation becomes

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-, t_k)) .$$

From the definition of the *a posteriori* state estimation error, it can be seen that

$$\mathbf{e}_k^+ = \mathbf{e}_k^- - \mathbf{K}_k [\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-, t_k)] .$$

Substituting the measurement equation given in Eq. (3.2) and applying a first-order Taylor series expansion yields

$$\mathbf{e}_k^+ = [\mathbf{I} - \mathbf{K}_k \mathbf{H}(\hat{\mathbf{x}}_k^-, t_k)] \mathbf{e}_k^- - \mathbf{K}_k \mathbf{v}_k , \tag{3.8}$$

where the measurement Jacobian,  $\mathbf{H}(\hat{\mathbf{x}}_k^-, t_k)$ , is defined as

$$\mathbf{H}(\hat{\mathbf{x}}_k^-, t_k) = \left[ \frac{\partial \mathbf{h}(\mathbf{x}_k, t_k)}{\partial \mathbf{x}_k} \right]_{\mathbf{x}_k = \hat{\mathbf{x}}_k^-} .$$

Computing the *a posteriori* state estimation error covariance by taking the expected value of Eq. (3.8) with its transpose then yields

$$\mathbf{P}_k^+ = [\mathbf{I} - \mathbf{K}_k \mathbf{H}(\hat{\mathbf{x}}_k^-, t_k)] \mathbf{P}_k^- [\mathbf{I} - \mathbf{K}_k \mathbf{H}(\hat{\mathbf{x}}_k^-, t_k)]^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T, \quad (3.9)$$

where it has been assumed that the *a priori* state estimation error and the measurement noise are uncorrelated. Eq. (3.9) is the well-known Joseph form of the covariance update, which is valid for any linear gain  $\mathbf{K}_k$ .<sup>14,43</sup> Up to this point, no form has been given for the gain matrix,  $\mathbf{K}_k$ . The gain matrix  $\mathbf{K}_k$  is found such that the mean square of the *a posteriori* state estimation error is minimized. Let the performance index be

$$J = \mathbb{E} \{ (\mathbf{e}_k^+)^T (\mathbf{e}_k^+) \} = \mathbb{E} \{ \text{trace} \{ (\mathbf{e}_k^+) (\mathbf{e}_k^+)^T \} \} = \text{trace} \{ \mathbf{P}_k^+ \}.$$

The gain which renders the performance index stationary is

$$\frac{\partial J}{\partial \mathbf{K}_k} = -2 [\mathbf{H}(\hat{\mathbf{x}}_k^-, t_k) \mathbf{P}_k^-]^T + 2 \mathbf{K}_k [\mathbf{H}(\hat{\mathbf{x}}_k^-, t_k) \mathbf{P}_k^- \mathbf{H}^T(\hat{\mathbf{x}}_k^-, t_k) + \mathbf{R}_k] = \mathbf{0},$$

which yields the Kalman gain as

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T(\hat{\mathbf{x}}_k^-, t_k) [\mathbf{H}(\hat{\mathbf{x}}_k^-, t_k) \mathbf{P}_k^- \mathbf{H}^T(\hat{\mathbf{x}}_k^-, t_k) + \mathbf{R}_k]^{-1}.$$

## 3.2 The Unscented Kalman Filter

In developing the governing relationships for the EKF, extensive use of linearization for both the dynamics and the measurement equations was employed. However, the Kalman filtering paradigm does not require that the models be linear. In fact, all that is required is that we have consistent, minimum variance estimates

such that the distribution can be well-represented by its first two moments, that the measurement update be a linear scheme (that is it is a linear combination of the prior state estimate and the measurement information), and that accurate predictions of the first two moments can be made.<sup>70</sup> Under these three requirements, it can then be shown that the state estimate evolves as

$$\dot{\hat{\mathbf{x}}}(t) = \mathbf{E} \{ \mathbf{f}(\mathbf{x}(t), t) \} ,$$

such that the second central moment can be computed via

$$\mathbf{P}_k^- = \mathbf{E} \{ (\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T \} .$$

Then, when measurement updates are considered, we have the predicted measurement as

$$\hat{\mathbf{y}}_k^- = \mathbf{E} \{ \mathbf{h}(\mathbf{x}_k, t_k) \} ,$$

which allows the state estimate and the covariance to be updated (assuming a linear scheme for the update), yielding

$$\begin{aligned} \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k^-) \\ \mathbf{P}_k^+ &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_y \mathbf{K}_k^T , \end{aligned}$$

where the Kalman gain is given by

$$\begin{aligned} \mathbf{K}_k &= \mathbf{E} \{ (\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{y}_k - \hat{\mathbf{y}}_k^-)^T \} \mathbf{E} \{ (\mathbf{y}_k - \hat{\mathbf{y}}_k^-)(\mathbf{y}_k - \hat{\mathbf{y}}_k^-)^T \}^{-1} \\ &= \mathbf{P}_{xy} \mathbf{P}_y^{-1} . \end{aligned}$$

Note that if the linearization procedures described in the development of the EKF are implemented in the above relationships, then we recover the EKF. However, it is not necessary to consider linearizations. One such method which forgoes linearization in favor of a more accurate computation is the unscented Kalman filter (UKF).

### 3.2.1 The Unscented Transform

Consider a nonlinear function of the form

$$\mathbf{z} = \mathbf{g}(\mathbf{x}),$$

where  $\mathbf{x}$  is described by a known mean and covariance, respectively  $\mathbf{m}_x$  and  $\mathbf{P}_x$ . The unscented transform (UT) seeks to approximate the transformation of the mean and covariance of the output,  $\mathbf{z}$ , which are denoted by  $\mathbf{m}_z$  and  $\mathbf{P}_z$ .

Whereas linearization methods utilize a first-order Taylor series expansion to approximate the transformation of the mean and covariance through a nonlinear function, the UT approaches the problem under the philosophy that it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function.<sup>67</sup> To approximate the probability distribution, the UT considers a set of deterministically chosen weighted sigma-points which are selected such that  $\mathbf{m}_x$  and  $\mathbf{P}_x$  are exactly captured by the sigma-points. The sigma-points are then applied as inputs to the nonlinear function to yield nonlinearly transformed sigma-points, which can then be used to approximate a nonlinear transformation of the output mean and covariance,  $\mathbf{m}_z$  and  $\mathbf{P}_z$ .

Let the set of sigma-points be denoted by the  $K$  values of  $\mathbf{x}_i$  and the associated

weights by  $w_i$  where  $i \in \{1, \dots, K\}$  and  $\sum_{i=1}^K w_i = 1$ . Then, the set of transformed sigma-points are given by

$$\mathbf{Z}_i = \mathbf{g}(\mathbf{X}_i) \quad \forall \quad i \in \{1, \dots, K\},$$

which are then used to compute the transformed mean and covariance as

$$\begin{aligned} \mathbf{m}_z &= \sum_{i=1}^K w_i \mathbf{Z}_i \\ \mathbf{P}_z &= \sum_{i=1}^K w_i (\mathbf{Z}_i - \mathbf{m}_z)(\mathbf{Z}_i - \mathbf{m}_z)^T. \end{aligned}$$

Additionally, the cross-covariance between the input and the output can be computed, if desired, as

$$\mathbf{P}_{xz} = \sum_{i=1}^K w_i (\mathbf{X}_i - \mathbf{m}_x)(\mathbf{Z}_i - \mathbf{m}_z)^T.$$

Any selection of sigma-points that exactly describes the input mean and covariance guarantees that the transformed mean and covariance is correctly calculated to second order.<sup>25</sup>

Many possibilities exist for the selection of the sigma-points and the associated weights, such as the simplex set, symmetric set, symmetric extended set, among others.<sup>25</sup> We restrict our attention to the symmetric set, which is given by the set of  $K = 2n$  points chosen as

$$\begin{aligned} w_i &= \frac{1}{2n} \\ \mathbf{X}_i &= \mathbf{m}_x + \sqrt{n} \mathbf{s}_{x,i} \\ w_{i+n} &= \frac{1}{2n} \\ \mathbf{X}_{i+n} &= \mathbf{m}_x - \sqrt{n} \mathbf{s}_{x,i}, \end{aligned}$$

for  $i \in \{1, \dots, n\}$ , where  $n$  is the dimension of the input  $\mathbf{x}$ ,  $\mathbf{S}_x$  is a square-root factor of  $\mathbf{P}_x$  such that  $\mathbf{P}_x = \mathbf{S}_x \mathbf{S}_x^T$ , and  $\mathbf{s}_{x,i}$  is the  $i^{\text{th}}$  column of  $\mathbf{S}_x$ . It is easily verified that the symmetric set of sigma-points matches the mean and covariance of  $\mathbf{x}$ , that is

$$\begin{aligned}\mathbf{m}_x &= \sum_{i=1}^K w_i \boldsymbol{\chi}_i \\ \mathbf{P}_x &= \sum_{i=1}^K w_i (\boldsymbol{\chi}_i - \mathbf{m}_x)(\boldsymbol{\chi}_i - \mathbf{m}_x)^T.\end{aligned}$$

An appealing aspect of using the symmetric set of sigma-points is that no tuning parameter is required in the selection of the points. The same cannot be said for other sigma-point selection schemes. While the ability to tune the sigma-points to achieve better accuracy is sometimes desired, there are only heuristic guidelines for selecting the tuning parameters, and small changes in tuning parameters can significantly alter the computation of the transformed mean and covariance, which leads to uncertainty in the choice of an “optimal” tuning. However, while the symmetric sigma-point set does not require tuning, the sigma-points lie on the  $\sqrt{n}^{\text{th}}$  covariance contour and therefore tend to move far away from the mean as  $n$  grows large.<sup>67</sup>

### 3.2.2 Propagation

In order to apply the UT to the forward propagation of the estimate and covariance, the first step is to determine the square-root factor of the covariance matrix, that is to find  $\mathbf{S}_{k-1}$  such that  $\mathbf{P}_{k-1} = \mathbf{S}_{k-1} \mathbf{S}_{k-1}^T$ , which can be readily accomplished via a Cholesky factorization. Once the square-root factor is determined, the columns

of the square-root factor, given by

$$\mathbf{S}_{k-1} = [\mathbf{s}_{1,k-1} \ \dots \ \mathbf{s}_{n,k-1}]$$

are used to determine the set of  $K = 2n$  sigma-points which make up the symmetric sigma-point set, such that for  $i \in \{1, \dots, n\}$

$$\begin{aligned}\boldsymbol{\chi}_{i,k-1} &= \hat{\mathbf{x}}_{k-1} + \sqrt{n}\mathbf{s}_{i,k-1} \\ \boldsymbol{\chi}_{i+n,k-1} &= \hat{\mathbf{x}}_{k-1} - \sqrt{n}\mathbf{s}_{i,k-1}.\end{aligned}$$

Associated with each sigma-point is a corresponding weight  $w_i$ . For the symmetric sigma-point set, the weights are given by  $w_i = 1/2n$  for all of the sigma-points. Each sigma-point is then numerically integrated through the nonlinear dynamics for  $t \in [t_{k-1} \ t_k]$  with an initial condition of  $\boldsymbol{\chi}_i(t_{k-1}) = \boldsymbol{\chi}_{i,k-1}$ , that is

$$\dot{\boldsymbol{\chi}}_i(t) = \mathbf{f}(\boldsymbol{\chi}_i(t), t), \quad \boldsymbol{\chi}_i(t_{k-1}) = \boldsymbol{\chi}_{i,k-1}.$$

The final condition on the numerical integration of the sigma-points is then given for each sigma-point by  $\boldsymbol{\chi}_{i,k} = \boldsymbol{\chi}_i(t_k)$ , which can then be used to approximate the nonlinear transformation of the estimate and the covariance using

$$\begin{aligned}\hat{\mathbf{x}}_k^- &= \sum_{i=1}^K w_i \boldsymbol{\chi}_{i,k} \\ \mathbf{P}_k^- &= \sum_{i=1}^K w_i (\boldsymbol{\chi}_{i,k} - \hat{\mathbf{x}}_k^-)(\boldsymbol{\chi}_{i,k} - \hat{\mathbf{x}}_k^-)^T.\end{aligned}$$

### 3.2.3 Update

Using the propagated sigma-points at time  $t_k$ , the measurement-transformed sigma-points are given by evaluating the nonlinear measurement function  $\mathbf{h}(\cdot)$  at each

sigma-point, yielding the  $K$  transformed sigma-points as

$$\mathbf{y}_{i,k} = \mathbf{h}(\mathbf{x}_{i,k}, t_k).$$

The expected value of the measurement, the measurement covariance, and the cross-covariance are found in terms of the transformed sigma-points as

$$\begin{aligned}\hat{\mathbf{y}}_k^- &= \sum_{i=1}^k w_i \mathbf{y}_{i,k} \\ \mathbf{P}_y &= \sum_{i=1}^K w_i (\mathbf{y}_{i,k} - \hat{\mathbf{y}}_k^-)(\mathbf{y}_{i,k} - \hat{\mathbf{y}}_k^-)^T + \mathbf{R}_k \\ \mathbf{P}_{xy} &= \sum_{i=1}^K w_i (\mathbf{x}_{i,k} - \hat{\mathbf{x}}_k^-)(\mathbf{y}_{i,k} - \hat{\mathbf{y}}_k^-)^T.\end{aligned}$$

In terms of the measurement covariance and the cross-covariance between the state and the measurement, the Kalman gain is

$$\mathbf{K}_k = \mathbf{P}_{xy} \mathbf{P}_y^{-1},$$

and the associated updated state estimate and covariance are

$$\begin{aligned}\hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \hat{\mathbf{y}}_k^-) \\ \mathbf{P}_k^+ &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_y \mathbf{K}_k^T.\end{aligned}$$

### 3.3 The Gaussian Mixture Extended Kalman Filter

Underlying the development of both the EKF and the UKF is the assumption that the probability density function (pdf) is well described by the first two moments only. While this assumption often leads to acceptable approximate nonlinear filtering strategies, it can prove to be less than adequate in situations where the pdf is



not adequately described by only the first two moments or where the measurements are sparse, thereby limiting the application of both the EKF and UKF in situations requiring long time-scales between measurements. To address some of the shortcomings of the use of only the first two moments, the Gaussian mixture extended Kalman filter (GMEKF) assumes that the initial pdf is not adequately determined by the first two moments, but rather employs a Gaussian mixture model (GMM) of the initial pdf, given by

$$p(\mathbf{x}_0) = \sum_{\ell=1}^L \alpha_{\ell,0} p_g(\mathbf{x}_0; \hat{\mathbf{x}}_{\ell,0}, \mathbf{P}_{\ell,0}),$$

where the individual Gaussian component weights are constrained such that

$$0 \leq \alpha_{\ell} \leq 1 \quad \forall \quad \ell \in \{1, \dots, L\} \quad \text{and} \quad \sum_{\ell=1}^L \alpha_{\ell} = 1.$$

The parameters of the GMM, i.e.  $\hat{\mathbf{x}}_{\ell}$ ,  $\alpha_{\ell}$ , and  $\mathbf{P}_{\ell}$ , are typically interpreted as follows: the mean values represent the regions in the state-space where the majority of the probability mass is concentrated, the component weights represent the normalized probability of the localized probability mass, and the covariance matrices are used to limit the regions of the state-space about the means such that the each term in the GMM is effectively zero outside of a neighborhood of the mean.

### 3.3.1 Propagation

Now, consider the time-propagation of the pdf, that is, considering the time interval  $t \in [t_{k-1} \ t_k]$ , it is desired to approximate the conditional pdf at time  $t_k$  via

$$p(\mathbf{x}_k | \mathbf{y}^{k-1}) = \sum_{\ell=1}^L \alpha_{\ell,k}^- p_g(\mathbf{x}_k; \hat{\mathbf{x}}_{\ell,k}^-, \mathbf{P}_{\ell,k}^-)$$

based on the starting condition at time  $t_{k-1}$

$$p(\mathbf{x}_{k-1} | \mathbf{y}^{k-1}) = \sum_{\ell=1}^L \alpha_{\ell,k-1}^+ p_g(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{\ell,k-1}^+, \mathbf{P}_{\ell,k-1}^+).$$

The notation  $\mathbf{y}^{k-1}$  in the conditional pdf  $p(\mathbf{x}_k | \mathbf{y}^{k-1})$  is used to represent the collection of all measurement data up to and including time  $t_{k-1}$ , i.e.  $\mathbf{y}^{k-1} = \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{k-1}\}$ . This is the data upon which the conditioning of the state pdf is based, hence the terminology of the *conditional* pdf for  $p(\mathbf{x}_k | \mathbf{y}^{k-1})$ . Furthermore, the use of the superscript “−” indicates a value prior to an update, such that  $\alpha_{\ell,k}^-$  is the component weight at time  $t_k$  prior to incorporating measurement data. Similarly, the use of the superscript “+” indicates a value after an update, such that  $\alpha_{\ell,k-1}^+$  represents the component weight at time  $t_{k-1}$  after measurement data at that time was incorporated.

Recalling that the covariance matrices of the components are used to limit the region of the state-space about which each component is valid, the dynamical system local to each component may be approximated via a first-order Taylor series expansion, thereby allowing the implementation of an EKF propagation scheme for each component while holding the component weights equal across the time-step. That is, by integrating

$$\begin{aligned} \dot{\alpha}_\ell(t) &= 0, \quad \alpha_\ell(t_{k-1}) = \alpha_{\ell,k-1}^+ \\ \dot{\hat{\mathbf{x}}}_\ell(t) &= \mathbf{f}(\hat{\mathbf{x}}_\ell(t), t), \quad \hat{\mathbf{x}}_\ell(t_{k-1}) = \hat{\mathbf{x}}_{\ell,k-1}^+ \\ \dot{\Phi}_\ell(t, t_{k-1}) &= \mathbf{F}(\hat{\mathbf{x}}_\ell(t), t) \Phi_\ell(t, t_{k-1}), \quad \Phi_\ell(t_{k-1}, t_{k-1}) = \mathbf{I} \end{aligned}$$

on the interval  $t \in [t_{k-1}, t_k]$ , we have the weight, mean, and covariance for each

component at time  $t_k$  as

$$\begin{aligned}\alpha_{\ell,k}^- &= \alpha_\ell(t_{k-1}) \\ \hat{\mathbf{x}}_{\ell,k}^- &= \hat{\mathbf{x}}_\ell(t_k) \\ \mathbf{P}_{\ell,k}^- &= \mathbf{\Phi}_\ell(t_k, t_{k-1}) \mathbf{P}_\ell(t_{k-1}) \mathbf{\Phi}_\ell^T(t_k, t_{k-1}), \quad \mathbf{P}_\ell(t_{k-1}) = \mathbf{P}_{\ell,k-1}^+.\end{aligned}$$

It should be noted that the weights being constant across the time step relies on the assumption that the covariances of the components is small enough such that linearizing about the component means is a valid assumption.

### 3.3.2 Update

Following the approach of Alspach and Sorenson<sup>1</sup> and Anderson,<sup>2</sup> the *a posteriori* density is determined via application of Bayes rule to yield

$$p(\mathbf{x}_k | \mathbf{y}^k) = \frac{p(\mathbf{x}_k | \mathbf{y}^{k-1}) p(\mathbf{y}_k | \mathbf{x}_k)}{\int_{\mathbb{R}^n} p(\mathbf{x}_k | \mathbf{y}^{k-1}) p(\mathbf{y}_k | \mathbf{x}_k) d\mathbf{x}_k}. \quad (3.10)$$

It is useful to note that for  $\mathbf{h}(\cdot)$ ,  $\mathbf{R}_k$ ,  $\hat{\mathbf{x}}_k^-$ , and  $\mathbf{P}_k^-$  of appropriate dimensions with both  $\mathbf{R}_k$  and  $\mathbf{P}_k^-$  positive definite, then<sup>22,71</sup>

$$p_g(\mathbf{y}_k; \mathbf{h}(\mathbf{x}_k, t_k), \mathbf{R}_k) p_g(\mathbf{x}_k; \hat{\mathbf{x}}_k^-, \mathbf{P}_k^-) = \beta_k p_g(\mathbf{x}; \hat{\mathbf{x}}_k^+, \mathbf{P}_k^+), \quad (3.11)$$

where

$$\begin{aligned}\beta_k &= p_g(\mathbf{y}_k; \hat{\mathbf{y}}_k^-, P_y) \\ \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k^-) \\ \mathbf{P}_k^+ &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_y \mathbf{K}_k^T \\ \mathbf{K}_k &= \mathbf{P}_{xy} \mathbf{P}_y^{-1}\end{aligned}$$

and

$$\begin{aligned}\hat{\mathbf{y}}_k^- &= \mathbb{E} \{ \mathbf{h}(\mathbf{x}_k, t_k) \} \\ \mathbf{P}_{xy} &= \mathbb{E} \{ (\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{y}_k - \hat{\mathbf{y}}_k^-)^T \} \\ \mathbf{P}_y &= \mathbb{E} \{ (\mathbf{y}_k - \hat{\mathbf{y}}_k^-)(\mathbf{y}_k - \hat{\mathbf{y}}_k^-)^T \} + \mathbf{R}_k.\end{aligned}$$

By linearizing the measurement about  $\hat{\mathbf{x}}_k^-$ , it follows from the development of the EKF equations that

$$\begin{aligned}\hat{\mathbf{y}}_k^- &= \mathbf{h}(\hat{\mathbf{x}}_k^-, t_k) \\ \mathbf{P}_{xy} &= \mathbf{P}_k^- \mathbf{H}^T(\hat{\mathbf{x}}_k^-, t_k) \\ \mathbf{P}_y &= \mathbf{H}(\hat{\mathbf{x}}_k^-, t_k) \mathbf{P}_k^- \mathbf{H}^T(\hat{\mathbf{x}}_k^-, t_k) + \mathbf{R}_k,\end{aligned}$$

and therefore, the parameters of the right-hand side of Eq. (3.11) can be written as

$$\begin{aligned}\beta_k &= p_g(\mathbf{y}_k; \mathbf{h}(\hat{\mathbf{x}}_k^-, t_k), \mathbf{H}(\hat{\mathbf{x}}_k^-, t_k) \mathbf{P}_k^- \mathbf{H}^T(\hat{\mathbf{x}}_k^-, t_k) + \mathbf{R}_k) \\ \hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-, t_k)) \\ \mathbf{P}_k^+ &= [\mathbf{I} - \mathbf{K}_k \mathbf{H}(\hat{\mathbf{x}}_k^-, t_k)] \mathbf{P}_k^- [\mathbf{I} - \mathbf{K}_k \mathbf{H}(\hat{\mathbf{x}}_k^-, t_k)]^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \\ \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}^T(\hat{\mathbf{x}}_k^-, t_k) [\mathbf{H}(\hat{\mathbf{x}}_k^-, t_k) \mathbf{P}_k^- \mathbf{H}^T(\hat{\mathbf{x}}_k^-, t_k) + \mathbf{R}_k]^{-1}.\end{aligned}$$

From the propagation stage, the conditional pdf of the state given the previous measurements is given by

$$p(\mathbf{x}_k | \mathbf{y}^{k-1}) = \sum_{\ell=1}^L \alpha_{\ell,k}^- p_g(\mathbf{x}_k; \hat{\mathbf{x}}_{\ell,k}^-, \mathbf{P}_{\ell,k}^-). \quad (3.12)$$

Furthermore, the conditional pdf of the current measurement given the state at time  $t_k$  is assumed to be Gaussian, such that

$$p(\mathbf{y}_k | \mathbf{x}_k) = p_g(\mathbf{y}_k; \mathbf{h}(\mathbf{x}_k, t_k), \mathbf{R}_k). \quad (3.13)$$

Note that the Gaussian assumption of  $p(\mathbf{y}_k | \mathbf{x}_k)$  is not necessary. As previously discussed, any pdf may be approximated to arbitrary precision by a GMM, and therefore the Gaussian assumption can be relaxed. However, for notational simplicity, this generalization is not employed, and we instead proceed under the assumption that  $p(\mathbf{y}_k | \mathbf{x}_k)$  is Gaussian. Looking now at the numerator of Eq. (3.10) and substituting for  $p(\mathbf{x}_k | \mathbf{y}^{k-1})$  from Eq. (3.12) and for  $p(\mathbf{y}_k | \mathbf{x}_k)$  from Eq. (3.13), it is seen that

$$\begin{aligned} p(\mathbf{x}_k | \mathbf{y}^{k-1})p(\mathbf{y}_k | \mathbf{x}_k) &= \left[ \sum_{\ell=1}^L \alpha_{\ell,k}^- p_g(\mathbf{x}_k; \hat{\mathbf{x}}_{\ell,k}^-, \mathbf{P}_{\ell,k}^-) \right] p_g(\mathbf{y}_k; \mathbf{h}(\mathbf{x}_k, t_k), \mathbf{R}_k) \\ &= \sum_{\ell=1}^L \alpha_{\ell,k}^- p_g(\mathbf{x}_k; \hat{\mathbf{x}}_{\ell,k}^-, \mathbf{P}_{\ell,k}^-) p_g(\mathbf{y}_k; \mathbf{h}(\mathbf{x}_k, t_k), \mathbf{R}_k). \end{aligned}$$

Therefore, applying Eq. (3.11) to each of the components to update the component means and covariances, yields the final form of the numerator of Eq. (3.10) as

$$p(\mathbf{x}_k | \mathbf{y}^{k-1})p(\mathbf{y}_k | \mathbf{x}_k) = \sum_{\ell=1}^L \beta_{\ell,k} \alpha_{\ell,k}^- p_g(\mathbf{x}_k; \hat{\mathbf{x}}_{\ell,k}^+, \mathbf{P}_{\ell,k}^+), \quad (3.14)$$

where

$$\begin{aligned} \beta_{\ell,k} &= p_g(\mathbf{y}_k; \mathbf{h}(\hat{\mathbf{x}}_{\ell,k}^-, t_k), \mathbf{H}(\hat{\mathbf{x}}_{\ell,k}^-, t_k) \mathbf{P}_{\ell,k}^- \mathbf{H}^T(\hat{\mathbf{x}}_{\ell,k}^-, t_k) + \mathbf{R}_k) \\ \hat{\mathbf{x}}_{\ell,k}^+ &= \hat{\mathbf{x}}_{\ell,k}^- + \mathbf{K}_{\ell,k}(\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{\ell,k}^-, t_k)) \\ \mathbf{P}_{\ell,k}^+ &= [\mathbf{I} - \mathbf{K}_{\ell,k} \mathbf{H}(\hat{\mathbf{x}}_{\ell,k}^-, t_k)] \mathbf{P}_{\ell,k}^- [\mathbf{I} - \mathbf{K}_{\ell,k} \mathbf{H}(\hat{\mathbf{x}}_{\ell,k}^-, t_k)]^T + \mathbf{K}_{\ell,k} \mathbf{R}_k \mathbf{K}_{\ell,k}^T \\ \mathbf{K}_{\ell,k} &= \mathbf{P}_{\ell,k}^- \mathbf{H}^T(\hat{\mathbf{x}}_{\ell,k}^-, t_k) [\mathbf{H}(\hat{\mathbf{x}}_{\ell,k}^-, t_k) \mathbf{P}_{\ell,k}^- \mathbf{H}^T(\hat{\mathbf{x}}_{\ell,k}^-, t_k) + \mathbf{R}_k]^{-1}. \end{aligned}$$

Now, consider the denominator of Eq. (3.10), which is given by the integral of

Eq. (3.14), yielding

$$\begin{aligned}
\int_{\mathbb{R}^n} p(\mathbf{x}_k | \mathbf{y}^{k-1}) p(\mathbf{y}_k | \mathbf{x}_k) d\mathbf{x}_k &= \int_{\mathbb{R}^n} \sum_{\ell=1}^L \beta_{\ell,k} \alpha_{\ell,k}^- p_g(\mathbf{x}_k; \hat{\mathbf{x}}_{\ell,k}^+, \mathbf{P}_{\ell,k}^+) d\mathbf{x}_k \\
&= \sum_{\ell=1}^L \beta_{\ell,k} \alpha_{\ell,k}^- \int_{\mathbb{R}^n} p_g(\mathbf{x}_k; \hat{\mathbf{x}}_{\ell,k}^+, \mathbf{P}_{\ell,k}^+) d\mathbf{x}_k \\
&= \sum_{\ell=1}^L \beta_{\ell,k} \alpha_{\ell,k}^-, \tag{3.15}
\end{aligned}$$

where the fact that the integral of any valid pdf over its entire domain is one has been used. Combining the results of Eqs. (3.14) and (3.15), therefore gives  $p(\mathbf{x}_k | \mathbf{y}^k)$  as

$$p(\mathbf{x}_k | \mathbf{y}^k) = \frac{1}{\sum_{\ell=1}^L \beta_{\ell,k} \alpha_{\ell,k}^-} \sum_{\ell=1}^L \beta_{\ell,k} \alpha_{\ell,k}^- p_g(\mathbf{x}_k; \hat{\mathbf{x}}_{\ell,k}^+, \mathbf{P}_{\ell,k}^+),$$

which can be rearranged yielding

$$p(\mathbf{x}_k | \mathbf{y}^k) = \sum_{\ell=1}^L \alpha_{\ell,k}^+ p_g(\mathbf{x}_k; \hat{\mathbf{x}}_{\ell,k}^+, \mathbf{P}_{\ell,k}^+),$$

where the *a posteriori* component weights are now given by

$$\alpha_{\ell,k}^+ = \frac{\beta_{\ell,k} \alpha_{\ell,k}^-}{\sum_{j=1}^L \beta_{j,k} \alpha_{j,k}^-}.$$

### 3.4 The Gaussian Mixture Unscented Kalman Filter

The next step in extending the application of the general method of Gaussian mixture approximations is to apply the UKF philosophy to the Gaussian mixture approximation method detailed in the development of the GMEKF. In a similar fashion to the development of the GMEKF, the Gaussian mixture unscented Kalman

filter (GMUKF) assumes that the initial pdf is not adequately determined by the first two moments, but rather employs a GMM of the initial pdf given by

$$p(\mathbf{x}_0) = \sum_{\ell=1}^L \alpha_{\ell,0} p_g(\mathbf{x}_0; \hat{\mathbf{x}}_{\ell,0}, \mathbf{P}_{\ell,0}),$$

where the individual Gaussian component weights are constrained such that

$$0 \leq \alpha_\ell \leq 1 \quad \forall \ell \in \{1, \dots, L\} \quad \text{and} \quad \sum_{\ell=1}^L \alpha_\ell = 1.$$

The parameters of the GMM, i.e.  $\hat{\mathbf{x}}_\ell$ ,  $\alpha_\ell$ , and  $\mathbf{P}_\ell$ , are interpreted as follows: the mean values represent the regions in the state-space where the majority of the probability mass is concentrated, the component weights represent the normalized probability of the localized probability mass, and the covariance matrices are used to limit the regions of the state-space about the means such that the each term in the GMM is effectively zero outside of a neighborhood of the mean.

### 3.4.1 Propagation

Consider the time-propagation of the pdf. Given the time interval  $t \in [t_{k-1}, t_k]$ , it is desired to approximate the conditional pdf at time  $t_k$  via

$$p(\mathbf{x}_k | \mathbf{y}^{k-1}) = \sum_{\ell=1}^L \alpha_{\ell,k}^- p_g(\mathbf{x}_k; \hat{\mathbf{x}}_{\ell,k}^-, \mathbf{P}_{\ell,k}^-) \quad (3.16)$$

based on the starting condition at time  $t_{k-1}$

$$p(\mathbf{x}_{k-1} | \mathbf{y}^{k-1}) = \sum_{\ell=1}^L \alpha_{\ell,k-1}^+ p_g(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{\ell,k-1}^+, \mathbf{P}_{\ell,k-1}^+).$$

Recalling that the covariance matrices of the components are used to limit the region of the state-space about which each component is valid, the dynamical system local to

each component may be approximated via the UT, thereby allowing the implementation of a UKF propagation scheme for each component while holding the component weights equal across the time-step. To apply the UT to the forward propagation of each component mean and covariance, first determine the square-root factor of the component covariance matrix, that is, find  $\mathbf{S}_{\ell,k-1}$  such that  $\mathbf{P}_{\ell,k-1}^+ = \mathbf{S}_{\ell,k-1} \mathbf{S}_{\ell,k-1}^T$ , which can be readily accomplished via a Cholesky factorization. Once the square-root factor is determined, the columns of the square-root factor, given by

$$\mathbf{S}_{\ell,k-1} = [\mathbf{s}_{\ell,1,k-1} \quad \dots \quad \mathbf{s}_{\ell,n,k-1}]$$

are used to determine the set of  $K = 2n$  sigma-points which make up the symmetric sigma-point set, such that for  $i \in \{1, \dots, n\}$ , each component sigma-points are given by

$$\begin{aligned} \mathbf{x}_{\ell,i,k-1} &= \hat{\mathbf{x}}_{\ell,k-1}^+ + \sqrt{n} \mathbf{s}_{\ell,i,k-1} \\ \mathbf{x}_{\ell,i+n,k-1} &= \hat{\mathbf{x}}_{\ell,k-1}^+ - \sqrt{n} \mathbf{s}_{\ell,i,k-1} . \end{aligned}$$

Associated with each sigma-point is a corresponding weight  $w_i$ . For the symmetric sigma-point set, the weights are given by  $w_i = 1/2n$  for all sigma-points. Each sigma-point is then numerically integrated through the nonlinear dynamics for  $t \in [t_{k-1}, t_k]$  with an initial condition of  $\mathbf{x}_{\ell,i}(t_{k-1}) = \mathbf{x}_{\ell,i,k-1}$ , that is

$$\dot{\mathbf{x}}_{\ell,i}(t) = \mathbf{f}(\mathbf{x}_{\ell,i}(t), t), \quad \mathbf{x}_{\ell,i}(t_{k-1}) = \mathbf{x}_{\ell,i,k-1} .$$

Additionally, to each component is the associated weight  $\alpha_\ell$ , which is held constant across each time step, or for  $t \in [t_{k-1}, t_k]$ , we have

$$\dot{\alpha}_\ell(t) = 0, \quad \alpha_\ell(t_{k-1}) = \alpha_{\ell,k-1}^+,$$



with a final condition of  $\alpha_{\ell,k}^- = \alpha_\ell(t_k)$ . The final condition on the numerical integration of the sigma-points is then given for each sigma-point by  $\mathbf{x}_{\ell,i,k} = \mathbf{x}_{\ell,i}(t_k)$ , which can then be used to approximate the nonlinear transformation of the component means and covariances using

$$\begin{aligned}\hat{\mathbf{x}}_{\ell,k}^- &= \sum_{i=1}^K w_i \mathbf{x}_{\ell,i,k} \\ \mathbf{P}_{\ell,k}^- &= \sum_{i=1}^K w_i (\mathbf{x}_{\ell,i,k} - \hat{\mathbf{x}}_{\ell,k}^-)(\mathbf{x}_{\ell,i,k} - \hat{\mathbf{x}}_{\ell,k}^-)^T.\end{aligned}$$

Combining the component means and covariance with the final condition on the numerical integration of the component weights, the *a priori* GMM pdf can be evaluated with Eq. (3.16).

### 3.4.2 Update

In the same manner as considered for the GMEKF, the *a posteriori* pdf is found by considering the composition of the *a priori* pdf and the measurement pdf at time  $t_k$ , and then normalizing the result, yielding

$$p(\mathbf{x}_k | \mathbf{y}^k) = \frac{p(\mathbf{x}_k | \mathbf{y}^{k-1})p(\mathbf{y}_k | \mathbf{x}_k)}{\int_{\mathbb{R}^n} p(\mathbf{x}_k | \mathbf{y}^{k-1})p(\mathbf{y}_k | \mathbf{x}_k)d\mathbf{x}_k}. \quad (3.17)$$

From the propagation stage, the conditional pdf of the state given the previous measurements is given by

$$p(\mathbf{x}_k | \mathbf{y}^{k-1}) = \sum_{\ell=1}^L \alpha_{\ell,k}^- p_g(\mathbf{x}_k; \hat{\mathbf{x}}_{\ell,k}^-, \mathbf{P}_{\ell,k}^-), \quad (3.18)$$

where, the mean and covariance for each component are fully captured by the propagated sigma-points,  $\mathbf{x}_{\ell,i,k}$ . Furthermore, the conditional pdf of the current measure-

ment given the state at time  $t_k$  is assumed to be Gaussian, such that

$$p(\mathbf{y}_k | \mathbf{x}_k) = p_g(\mathbf{y}_k; \mathbf{h}(\mathbf{x}_k, t_k), \mathbf{R}_k). \quad (3.19)$$

The Gaussian assumption of  $p(\mathbf{y}_k | \mathbf{x}_k)$  is not necessary (as discussed before); however, for notational simplicity we proceed under the assumption that  $p(\mathbf{y}_k | \mathbf{x}_k)$  is Gaussian. Considering the numerator of Eq. (3.17) and substituting for  $p(\mathbf{x}_k | \mathbf{y}^{k-1})$  from Eq. (3.18) and for  $p(\mathbf{y}_k | \mathbf{x}_k)$  from Eq. (3.19) yields

$$p(\mathbf{x}_k | \mathbf{y}^{k-1})p(\mathbf{y}_k | \mathbf{x}_k) = \sum_{\ell=1}^L \alpha_{\ell,k}^- p_g(\mathbf{x}_k; \hat{\mathbf{x}}_{\ell,k}^-, \mathbf{P}_{\ell,k}^-) p_g(\mathbf{y}_k; \mathbf{h}(\mathbf{x}_k, t_k), \mathbf{R}_k).$$

Therefore, by applying Eq. (3.11) to each of the components to update the component means and covariances, the final form of the numerator of Eq. (3.17) is

$$p(\mathbf{x}_k | \mathbf{y}^{k-1})p(\mathbf{y}_k | \mathbf{x}_k) = \sum_{\ell=1}^L \beta_{\ell,k} \alpha_{\ell,k}^- p_g(\mathbf{x}_k; \hat{\mathbf{x}}_{\ell,k}^+, \mathbf{P}_{\ell,k}^+), \quad (3.20)$$

where

$$\beta_{\ell,k} = p_g(\mathbf{y}_k; \hat{\mathbf{y}}_{\ell,k}^-, P_{\ell,y}) \quad (3.21a)$$

$$\hat{\mathbf{x}}_{\ell,k}^+ = \hat{\mathbf{x}}_{\ell,k}^- + \mathbf{K}_{\ell,k}(\mathbf{y}_k - \hat{\mathbf{y}}_{\ell,k}^-) \quad (3.21b)$$

$$\mathbf{P}_{\ell,k}^+ = \mathbf{P}_{\ell,k}^- - \mathbf{K}_{\ell,k} \mathbf{P}_{\ell,y} \mathbf{K}_{\ell,k}^T \quad (3.21c)$$

$$\mathbf{K}_{\ell,k} = \mathbf{P}_{\ell,xy} \mathbf{P}_{\ell,y}^{-1}. \quad (3.21d)$$

To compute the expected value of  $\mathbf{y}_k$  as well as its covariance and the cross covariance between the *a priori* state and the measurement, the UT is utilized. In doing so, the first step is to compute the measurement-transformed sigma-points for each component as

$$\mathbf{y}_{\ell,i,k} = \mathbf{h}(\mathbf{x}_{\ell,i,k}, t_k).$$

Then, for each component,  $\hat{\mathbf{y}}_{\ell,k}^-$ ,  $\mathbf{P}_{\ell,y}$ , and  $\mathbf{P}_{\ell,xy}$  are found to be

$$\begin{aligned}\hat{\mathbf{y}}_{\ell,k}^- &= \sum_{i=1}^k w_i \mathcal{Y}_{\ell,i,k} \\ \mathbf{P}_{\ell,y} &= \sum_{i=1}^k w_i (\mathcal{Y}_{\ell,i,k} - \hat{\mathbf{y}}_{\ell,k}^-)(\mathcal{Y}_{\ell,i,k} - \hat{\mathbf{y}}_{\ell,k}^-)^T + \mathbf{R}_k \\ \mathbf{P}_{\ell,xy} &= \sum_{i=1}^k w_i (\mathcal{X}_{\ell,i,k} - \hat{\mathbf{x}}_{\ell,k}^-)(\mathcal{Y}_{\ell,i,k} - \hat{\mathbf{y}}_{\ell,k}^-)^T.\end{aligned}$$

The component means and covariances can then be updated via Eqs. (3.21). To determine the update rule for the component weights, we must complete the development of Eq. (3.17). To do this, consider the denominator of Eq. (3.17), which is given by the integral of Eq. (3.20), yielding

$$\int_{\mathbb{R}^n} p(\mathbf{x}_k | \mathbf{y}^{k-1}) p(\mathbf{y}_k | \mathbf{x}_k) d\mathbf{x}_k = \sum_{\ell=1}^L \beta_{\ell,k} \alpha_{\ell,k}^- . \quad (3.22)$$

Combining the results of Eqs. (3.20) and (3.22) yields  $p(\mathbf{x}_k | \mathbf{y}^k)$  as

$$p(\mathbf{x}_k | \mathbf{y}^k) = \frac{1}{\sum_{\ell=1}^L \beta_{\ell,k} \alpha_{\ell,k}^-} \sum_{\ell=1}^L \beta_{\ell,k} \alpha_{\ell,k}^- p_g(\mathbf{x}_k; \hat{\mathbf{x}}_{\ell,k}^+, \mathbf{P}_{\ell,k}^+),$$

which can be rearranged to give

$$p(\mathbf{x}_k | \mathbf{y}^k) = \sum_{\ell=1}^L \alpha_{\ell,k}^+ p_g(\mathbf{x}_k; \hat{\mathbf{x}}_{\ell,k}^+, \mathbf{P}_{\ell,k}^+),$$

where the *a posteriori* component weights are now given by

$$\alpha_{\ell,k}^+ = \frac{\beta_{\ell,k} \alpha_{\ell,k}^-}{\sum_{j=1}^L \beta_{j,k} \alpha_{j,k}^-}.$$

### 3.5 The Splitting Gaussian Mixture Unscented Kalman Filter

The development of both the GMEKF and GMUKF filters relied on the weights of the components of the GMM pdf to be held constant over the propagation cycle. While this relies on linearization to be a valid approximation for the GMEKF, the GMUKF is able to relax the linearization constraint due to its implementation of the UT methodology. However, even with the relaxation of the linearization constraint, there is no method for either the GMEKF or GMUKF that allows for online adaptation of the GMM components. The splitting Gaussian mixture unscented Kalman filter (SGMUKF) approaches the problem of adapting the weights of the GMM pdf by monitoring nonlinearity during the propagation of the pdf, and using a splitting algorithm to increase the accuracy of linearization, thereby allowing the filter to modify the GMM components in such a way so as to avoid significant linearization errors.

#### 3.5.1 Detecting Nonlinearity during Propagation

Determination of the nonlinearity of a dynamical system has been previously investigated by Junkins<sup>27</sup> and then later by Park.<sup>48,49</sup> Junkins looked at nonlinearity of a dynamical system from the perspective of investigating the nonlinearity of different coordinate systems. By defining a nonlinearity index based on the state transition matrix, it was shown that different coordinate systems used in orbital mechanics (i.e. Cartesian, equinoctial, etc.) exhibit different levels of nonlinearity, thereby establishing that by choosing a particular coordinate system, the effects of nonlinearity can

be mitigated to some extent. Park extended Junkins' approach by considering the impact of higher-order dynamics by utilizing state transition tensors, which leads to an indication of how well a lower-order (but not necessarily linear) approximation of a nonlinear dynamical system follows the actual nonlinear dynamical system. In these cases, while the impact of nonlinearity is examined, the mitigation of nonlinearity is accomplished either via selection of an appropriate coordinate system or by inclusion of progressively higher-order terms.

One approach to detecting nonlinearity during propagation is to directly compare a linearized solution to a higher order solution. If the difference between these two solutions becomes large (in some sense), then the implication is that the higher-order terms are influencing the solutions. Consider then the implementation of an GMEKF and a GMUKF estimation scheme during propagation. If both filters are subjected to a linear dynamical system, then their component-by-component solutions will be identical (neglecting any computational and numerical differences). However, if the underlying dynamical system is nonlinear, then the solutions determined by the two propagation schemes will begin to depart. Since both the GMEKF and GMUKF operate on the mean and covariance of the components of a GMM distribution, the  $L_2$  or  $NL_2$  distance can be used to calculate how far apart the two solutions have become for a single component. Once this reaches a specified tolerance for a component, the propagation can be stopped, and the single component of the GMUKF distribution can be broken down into multiple smaller components by a splitting algorithm. The multiple smaller components then replace the single component, and the propagation resumes. The process can then be repeated on the new set of GMM components,

with component splitting occurring whenever the distance between the GMEKF and GMUKF prediction of each component becomes larger than a specified tolerance.

Unfortunately, the utilization of the  $L_2$  or  $NL_2$  distance based splitting technique requires the simultaneous implementation of both the GMEKF and GMUKF, which causes a higher computational demand than may be desired. To circumvent this computational demand, we now present a method which exploits a property derived from the differential entropy or Rényi entropy for linearized components that allows for the use of linearization-based methods to be avoided. Recall that the differential entropy for a Gaussian random variable  $\mathbf{x}$  is given by

$$H(\mathbf{x}) = \frac{1}{2} \log |2\pi e \mathbf{P}|, \quad (3.23)$$

or, in terms of the eigenvalues of the covariance matrix,

$$H(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^n \log(2\pi e \lambda_i), \quad (3.24)$$

where  $n$  is the dimension of the random variable  $\mathbf{x}$  and  $\lambda_i$  is the  $i^{\text{th}}$  eigenvalue of  $\mathbf{P}$ . Alternatively, the scaling terms inside of the logarithm of the determinant in Eq. (3.23) can be moved out, and  $H(\mathbf{x})$  can be written as

$$H(\mathbf{x}) = \frac{n}{2} \log(2\pi e) + \frac{1}{2} \log |\mathbf{P}|. \quad (3.25)$$

Similarly, the form of  $H(\mathbf{x})$  given by Eq. (3.24) which uses the eigenvalues of the covariance matrix can be rewritten as

$$H(\mathbf{x}) = \frac{n}{2} \log(2\pi e) + \frac{1}{2} \sum_{i=1}^n \log \lambda_i. \quad (3.26)$$

Consider now the temporal differentiation of the differential entropy. From the relationship<sup>3</sup>

$$\frac{d}{dt}\{|\mathbf{P}|\} = |\mathbf{P}| \operatorname{trace} \left\{ \mathbf{P}^{-1} \dot{\mathbf{P}} \right\} ,$$

it follows that the time derivative of Eq. (3.25) is given by

$$\dot{\mathbf{H}}(\mathbf{x}) = \frac{1}{2} \operatorname{trace} \left\{ \mathbf{P}^{-1} \dot{\mathbf{P}} \right\} , \quad (3.27)$$

where  $\dot{\mathbf{P}}$  is the temporal derivative of the covariance matrix. Starting from the form of the differential entropy in Eq. (3.26), it is also readily observed that the time-rate of the differential entropy is

$$\dot{\mathbf{H}}(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^n \frac{\dot{\lambda}_i}{\lambda_i} ,$$

where  $\dot{\lambda}_i$  is the temporal derivative of the  $i^{\text{th}}$  eigenvalue of  $\mathbf{P}$ , which can be computed in terms of  $\dot{\mathbf{P}}$  and the  $i^{\text{th}}$  eigenvector of the covariance matrix,  $\mathbf{e}_i$ , as<sup>7</sup>

$$\dot{\lambda}_i = \mathbf{e}_i^T \dot{\mathbf{P}} \mathbf{e}_i . \quad (3.28)$$

In the case of a linearized dynamical system, the covariance was shown to have the propagation equation of Eq. (3.5), i.e.

$$\mathbf{P}(t) = \mathbf{\Phi}(t, t_{k-1}) \mathbf{P}(t_{k-1}) \mathbf{\Phi}^T(t, t_{k-1}) ,$$

which, when differentiated with respect to time yields the covariance time rate given by

$$\dot{\mathbf{P}}(t) = \mathbf{F}(\hat{\mathbf{x}}(t), t) \mathbf{P}(t) + \mathbf{P}(t) \mathbf{F}^T(\hat{\mathbf{x}}(t), t) ,$$

which can then be substituted into Eq. (3.27) to yield the time rate of the differential entropy for a linearized dynamical system as

$$\begin{aligned}\dot{H}(\mathbf{x}) &= \frac{1}{2} \text{trace} \{ \mathbf{P}^{-1}(t) \mathbf{F}(\hat{\mathbf{x}}(t), t) \mathbf{P}(t) + \mathbf{P}^{-1}(t) \mathbf{P}(t) \mathbf{F}(\hat{\mathbf{x}}(t), t)^T \} \\ &= \frac{1}{2} \text{trace} \{ \mathbf{P}^{-1}(t) \mathbf{F}(\hat{\mathbf{x}}(t), t) \mathbf{P}(t) \} + \frac{1}{2} \text{trace} \{ \mathbf{F}^T(\hat{\mathbf{x}}(t), t) \} \\ &= \frac{1}{2} \text{trace} \{ \mathbf{F}(\hat{\mathbf{x}}(t), t) \} + \frac{1}{2} \text{trace} \{ \mathbf{F}^T(\hat{\mathbf{x}}(t), t) \} ,\end{aligned}$$

where the invariance under cyclic permutation property of the trace operator has been used to eliminate  $\mathbf{P}$  and  $\mathbf{P}^{-1}$  from the first term. Then, from the fact that  $\text{trace} \{ \mathbf{F}(\hat{\mathbf{x}}(t), t) \} = \text{trace} \{ \mathbf{F}^T(\hat{\mathbf{x}}(t), t) \}$ , it follows that

$$\dot{H}(\mathbf{x}) = \text{trace} \{ \mathbf{F}(\hat{\mathbf{x}}(t), t) \} , \quad (3.29)$$

which is a different form of a result given by Vallée.<sup>68,69</sup> Therefore, if a linearized dynamical system has the property that  $\text{trace} \{ \mathbf{F}(\hat{\mathbf{x}}(t), t) \} = 0$ , then the differential entropy is constant, that is

$$\dot{H}(\mathbf{x}) = 0 . \quad (3.30)$$

Since the differential entropy is in fact a specific case of the Rényi entropy, it stands to reason that a parallel result to Eq. (3.29) should exist. Recall that the Rényi entropy for a Gaussian random variable  $\mathbf{x}$  is given by

$$R_\kappa(\mathbf{x}) = \frac{1}{2} \log \left| 2\pi\kappa^{\frac{1}{\kappa-1}} \mathbf{P} \right| , \quad (3.31)$$

or, in terms of the eigenvalues of the covariance matrix,

$$R_\kappa(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^n \log(2\pi\kappa^{\frac{1}{\kappa-1}} \lambda_i) , \quad (3.32)$$



where  $n$  is the dimension of the random variable  $\mathbf{x}$  and  $\lambda_i$  is the  $i^{\text{th}}$  eigenvalue of  $\mathbf{P}$ . The Rényi entropy for a Gaussian pdf has the same form as the differential entropy for a Gaussian pdf, with only a change in the scaling of the covariance matrix inside of the logarithm of the determinant. Therefore, since the scaling of the covariance matrix is constant for a given Rényi entropy of order  $\kappa$ , the temporal derivative of the Rényi entropy is the same as that of the differential entropy, that is the time rate of change of Eq. (3.31) is given by

$$\dot{R}_\kappa(\mathbf{x}) = \frac{1}{2} \text{trace} \left\{ \mathbf{P}^{-1} \dot{\mathbf{P}} \right\}, \quad (3.33)$$

where  $\dot{\mathbf{P}}$  is the temporal derivative of the covariance matrix. In the same manner, but by using Eq. (3.32), it is readily established that

$$\dot{R}_\kappa(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^n \frac{\dot{\lambda}_i}{\lambda_i},$$

where  $\dot{\lambda}_i$  is the temporal derivative of the  $i^{\text{th}}$  eigenvalue of  $\mathbf{P}$ , which can be computed in terms of  $\dot{\mathbf{P}}$  and the  $i^{\text{th}}$  eigenvector of the covariance matrix,  $\mathbf{e}_i$ , via Eq. (3.28). Continuing along the same line of reasoning as followed for the case of differential entropy, and working from Eq. (3.33), it can be shown that for a linear system, the temporal derivative of the Rényi entropy is

$$\dot{R}_\kappa(\mathbf{x}) = \text{trace} \left\{ \mathbf{F}(\hat{\mathbf{x}}(t), t) \right\}, \quad (3.34)$$

which is also a different form of a result given by Vallée.<sup>69</sup> Therefore, if a linearized dynamical system has the property that  $\text{trace} \left\{ \mathbf{F}(\hat{\mathbf{x}}(t), t) \right\} = 0$ , then the Rényi entropy, like the differential entropy, is constant, that is

$$\dot{R}_\kappa(\mathbf{x}) = 0. \quad (3.35)$$

Having established the general relationships for the time rate of change of the differential entropy and Rényi entropy for a linearized dynamical system via Eqs. (3.29) and (3.34), respectively, the utilization of entropy for detection of nonlinearity is now discussed. The value of the entropy for a linearized system can be determined by numerically integrating either Eq. (3.29) for differential entropy or Eq. (3.34) for Rényi entropy with an appropriate initial condition, which requires only the evaluation of the trace of the linearized dynamics Jacobian. In parallel, a nonlinear implementation of the integration of the covariance matrix (such as is done in the UKF) can be considered, which allows a nonlinear determination of the differential entropy via Eq. (3.23) or the Rényi entropy via Eq. (3.31). Any deviation in the nonlinear determination of the entropy therefore indicates that nonlinearity is impacting the solution. This deviation can be detected by specifying a threshold and monitoring the difference between the linearized and nonlinear predictions of the entropy. In the special case that the linearized dynamical system has the property that the trace of the dynamics Jacobian is zero, the process is even simpler since the value of the entropy for the linearized system is constant (as demonstrated in Eq. (3.30) for the differential entropy and Eq. (3.35) for the Rényi entropy). Therefore, in this special case, only the nonlinear prediction of the entropy needs to be computed online and the deviation is determined by comparing against the entropy at some reference time (such as the initial time of the propagation). In either case, when the difference between the linearized-predicted entropy and the nonlinear computation of the entropy exceeds some threshold, nonlinearity has been detected in the propagation of the dynamical system.

### 3.5.2 Propagation

Consider the time-propagation of the pdf and consider the time interval  $t \in [t_{k-1}, t_k]$ . It is desired to approximate the conditional pdf at time  $t_k$  via

$$p(\mathbf{x}_k | \mathbf{y}^{k-1}) = \sum_{\ell=1}^{L'} \alpha_{\ell,k}^- p_g(\mathbf{x}_k; \hat{\mathbf{x}}_{\ell,k}^-, \mathbf{P}_{\ell,k}^-) \quad (3.36)$$

based on the starting condition at time  $t_{k-1}$

$$p(\mathbf{x}_{k-1} | \mathbf{y}^{k-1}) = \sum_{\ell=1}^L \alpha_{\ell,k-1}^+ p_g(\mathbf{x}_{k-1}; \hat{\mathbf{x}}_{\ell,k-1}^+, \mathbf{P}_{\ell,k-1}^+).$$

It should be noted that the number of components in  $p(\mathbf{x}_k | \mathbf{y}^{k-1})$ , given by  $L'$ , may now be different than the number of components in  $p(\mathbf{x}_{k-1} | \mathbf{y}^{k-1})$ , given by  $L$ . The change in the number of components reflects the ability of the SGMUKF to augment the number of components in the GMM in order to maintain linearity across each component during propagation.

In order to propagate the pdf forward, the first step is to determine the square-root factor of the component covariance matrices at time  $t_{k-1}$ , that is find  $\mathbf{S}_{\ell,k-1}$  such that  $\mathbf{P}_{\ell,k-1}^+ = \mathbf{S}_{\ell,k-1} \mathbf{S}_{\ell,k-1}^T$ , which can be readily accomplished via a Cholesky factorization. Once the square-root factor is determined, the columns of the square-root factor, given by

$$\mathbf{S}_{\ell,k-1} = [\mathbf{s}_{\ell,1,k-1} \ \cdots \ \mathbf{s}_{\ell,n,k-1}] \quad (3.37)$$

are used to determine the set of  $K = 2n$  sigma-points which make up the symmetric sigma-point set, such that for  $i \in \{1, \dots, n\}$ , each component sigma-points are given

by

$$\mathbf{x}_{\ell,i,k-1} = \hat{\mathbf{x}}_{\ell,k-1}^+ + \sqrt{n}\mathbf{s}_{\ell,i,k-1} \quad (3.38a)$$

$$\mathbf{x}_{\ell,i+n,k-1} = \hat{\mathbf{x}}_{\ell,k-1}^+ - \sqrt{n}\mathbf{s}_{\ell,i,k-1} . \quad (3.38b)$$

Associated with each sigma-point is a corresponding weight  $w_i$ . For the symmetric sigma-point set, the weights are given by  $w_i = 1/2n$  for all sigma-points.

Let the time  $t_s$  denote the time at which nonlinear effects (determined via the differential entropy) grow larger than a specified bound for one of the components of the GMM, thereby requiring a splitting step to be performed on the component. Furthermore, let  $t_{s-1}$  denote the previous time at which a splitting step was performed; initially, no splitting step has been performed, so  $t_{s-1}$  is initialized as  $t_{k-1}$ . Then, each sigma-point is numerically integrated through the nonlinear dynamics for  $t \in [t_{s-1}, t_s]$ , with an initial condition of  $\mathbf{x}_{\ell,i}(t_{s-1}) = \mathbf{x}_{\ell,i,s-1}$ , that is

$$\dot{\mathbf{x}}_{\ell,i}(t) = \mathbf{f}(\mathbf{x}_{\ell,i}(t), t), \quad \mathbf{x}_{\ell,i}(t_{s-1}) = \mathbf{x}_{\ell,i,s-1} . \quad (3.39)$$

Additionally, to each component is the associated weight  $\alpha_\ell$ , which is held constant across each time step, or for  $t \in [t_{s-1}, t_s]$

$$\dot{\alpha}_\ell(t) = 0, \quad \alpha_\ell(t_{s-1}) = \alpha_{\ell,s-1},$$

with a final condition of  $\alpha_{\ell,s} = \alpha_\ell(t_s)$ . The final condition on the numerical integration of the sigma-points for  $t \in [t_{s-1}, t_s]$  is then given for each sigma-point by  $\mathbf{x}_{\ell,i,s} = \mathbf{x}_{\ell,i}(t_s)$ , which can then be used to approximate the nonlinear transformation of the

component means and covariances using

$$\begin{aligned}\hat{\mathbf{x}}_{\ell,s} &= \sum_{i=1}^K w_i \mathbf{x}_{\ell,i,s} \\ \mathbf{P}_{\ell,s} &= \sum_{i=1}^K w_i (\mathbf{x}_{\ell,i,s} - \hat{\mathbf{x}}_{\ell,s})(\mathbf{x}_{\ell,i,s} - \hat{\mathbf{x}}_{\ell,s})^T.\end{aligned}$$

If  $t_s \neq t_k$ , then a splitting step is performed on the component for which nonlinearity was detected. That is, if nonlinearity was detected in the  $j^{\text{th}}$  component, then the  $j^{\text{th}}$  component is replaced by

$$\alpha_{j,s} p_g(\mathbf{x}; \hat{\mathbf{x}}_{j,s}, \mathbf{P}_{j,s}) \approx \sum_{r=1}^G \alpha_{r,s} p_g(\mathbf{x}; \hat{\mathbf{x}}_{r,s}, \mathbf{P}_{r,s}), \quad (3.40)$$

where the replacement component weights, means, and covariances are computed using the splitting algorithm given in Section 2.3. We then generate a set of sigma-points for the replacement components using Eqs. (3.37) and (3.38), and then return to Eq. (3.39) with  $L \leftarrow L + G - 1$  components, and continue until  $t_s = t_k$  is reached. Once  $t_s = t_k$ , the propagation step has been completed with  $L'$  components having weights  $\alpha_{\ell,k}^-$ , means  $\hat{\mathbf{x}}_{\ell,k}^-$ , and covariances  $\mathbf{P}_{\ell,k}^-$ , which allows the *a priori* GMM pdf to be evaluated via Eq. (3.36).

### 3.5.3 Update

The update stage of the SGMUKF remains unchanged from that of the GMUKF. Therefore, the derivation of the update stage is not presented, but for completeness, it is reviewed. The *a posteriori* pdf is found by considering the composition of the *a priori* pdf and the measurement pdf at time  $t_k$ , and then normalizing the result,

yielding

$$p(\mathbf{x}_k | \mathbf{y}^k) = \frac{p(\mathbf{x}_k | \mathbf{y}^{k-1})p(\mathbf{y}_k | \mathbf{x}_k)}{\int_{\mathbb{R}^n} p(\mathbf{x}_k | \mathbf{y}^{k-1})p(\mathbf{y}_k | \mathbf{x}_k)d\mathbf{x}_k}. \quad (3.41)$$

From the propagation stage previously discussed, the conditional pdf of the state given the previous measurements is given by

$$p(\mathbf{x}_k | \mathbf{y}^{k-1}) = \sum_{\ell=1}^L \alpha_{\ell,k}^- p_g(\mathbf{x}_k; \hat{\mathbf{x}}_{\ell,k}^-, \mathbf{P}_{\ell,k}^-), \quad (3.42)$$

where, for each component the mean and covariance are fully captured by the propagated sigma-points,  $\mathcal{X}_{\ell,i,k}$ . Furthermore, the conditional pdf of the current measurement given the state at time  $t_k$  is assumed to be Gaussian, such that

$$p(\mathbf{y}_k | \mathbf{x}_k) = p_g(\mathbf{y}_k; \mathbf{h}(\mathbf{x}_k, t_k), \mathbf{R}_k). \quad (3.43)$$

Given the *a priori* sigma-points, that is the sigma-points at the end of the propagation stage, the measurement transformed sigma-points are computed for each component by

$$\mathcal{Y}_{\ell,i,k} = \mathbf{h}(\mathcal{X}_{\ell,i,k}, t_k).$$

Then, for each component,  $\hat{\mathbf{y}}_{\ell,k}^-$ ,  $\mathbf{P}_{\ell,y}$ , and  $\mathbf{P}_{\ell,xy}$  are determined as

$$\begin{aligned} \hat{\mathbf{y}}_{\ell,k}^- &= \sum_{i=1}^k w_i \mathcal{Y}_{\ell,i,k} \\ \mathbf{P}_{\ell,y} &= \sum_{i=1}^K w_i (\mathcal{Y}_{\ell,i,k} - \hat{\mathbf{y}}_{\ell,k}^-)(\mathcal{Y}_{\ell,i,k} - \hat{\mathbf{y}}_{\ell,k}^-)^T + \mathbf{R}_k \\ \mathbf{P}_{\ell,xy} &= \sum_{i=1}^K w_i (\mathcal{X}_{\ell,i,k} - \hat{\mathbf{x}}_{\ell,k}^-)(\mathcal{Y}_{\ell,i,k} - \hat{\mathbf{y}}_{\ell,k}^-)^T. \end{aligned}$$

The Kalman gain and weight gain can then be directly obtained via

$$\mathbf{K}_{\ell,k} = \mathbf{P}_{\ell,xy} \mathbf{P}_{\ell,y}^{-1} \quad (3.44)$$

$$\beta_{\ell,k} = p_g(\mathbf{y}_k; \hat{\mathbf{y}}_{\ell,k}^-, P_{\ell,y}) \quad (3.45)$$

which then allows the weight, mean, and covariance of the components to be updated as

$$\alpha_{\ell,k}^+ = \frac{\beta_{\ell,k} \alpha_{\ell,k}^-}{\sum_{j=1}^L \beta_{j,k} \alpha_{j,k}^-} \quad (3.46)$$

$$\hat{\mathbf{x}}_{\ell,k}^+ = \hat{\mathbf{x}}_{\ell,k}^- + \mathbf{K}_{\ell,k} (\mathbf{y}_k - \hat{\mathbf{y}}_{\ell,k}^-) \quad (3.47)$$

$$\mathbf{P}_{\ell,k}^+ = \mathbf{P}_{\ell,k}^- - \mathbf{K}_{\ell,k} \mathbf{P}_{\ell,y} \mathbf{K}_{\ell,k}^T \quad (3.48)$$

## 3.6 Summary of Filter Algorithms

Having developed the governing equations for the EKF, UKF, GMEKF, and GMUKF, we now summarize each of the preceding algorithms.

### 3.6.1 EKF Algorithm

---

#### Algorithm 1. *EKF*

*System and Measurement Model*

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t)$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k, t_k) + \mathbf{v}_k$$

### Initialization

$$\begin{aligned}\hat{\mathbf{x}}_0 &= \mathbb{E} \{ \mathbf{x}(t_0) \} \\ \mathbf{P}_0 &= \mathbb{E} \{ (\mathbf{x}(t_0) - \hat{\mathbf{x}}_0)(\mathbf{x}(t_0) - \hat{\mathbf{x}}_0)^T \}\end{aligned}$$

### Definitions

$$\begin{aligned}\mathbf{F}(\hat{\mathbf{x}}(t), t) &= \left[ \frac{\partial \mathbf{f}(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)} \right]_{\mathbf{x}(t)=\hat{\mathbf{x}}(t)} \\ \mathbf{H}(\hat{\mathbf{x}}_k^-, t_k) &= \left[ \frac{\partial \mathbf{h}(\mathbf{x}_k, t_k)}{\partial \mathbf{x}_k} \right]_{\mathbf{x}_k=\hat{\mathbf{x}}_k^-}\end{aligned}$$

Propagation,  $t \in [t_{k-1}, t_k]$

1. Propagate mean and state transition matrix through the dynamics

$$\begin{aligned}\dot{\hat{\mathbf{x}}}(t) &= \mathbf{f}(\hat{\mathbf{x}}(t), t), \quad \hat{\mathbf{x}}(t_{k-1}) = \hat{\mathbf{x}}_{k-1}^+, \quad \hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}(t_k) \\ \dot{\Phi}(t, t_{k-1}) &= \mathbf{F}(\hat{\mathbf{x}}(t), t) \Phi(t, t_{k-1}), \quad \Phi(t_{k-1}, t_{k-1}) = \mathbf{I}\end{aligned}$$

2. Calculate propagated covariance

$$\mathbf{P}(t) = \Phi(t, t_{k-1}) \mathbf{P}(t_{k-1}) \Phi^T(t, t_{k-1}), \quad \mathbf{P}(t_{k-1}) = \mathbf{P}_{k-1}^+, \quad \mathbf{P}_k^- = \mathbf{P}(t_k)$$

Update,  $t_k \quad \forall \quad k = 1, 2, \dots$

1. Compute Kalman gain

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T(\hat{\mathbf{x}}_k^-, t_k) \left[ \mathbf{H}(\hat{\mathbf{x}}_k^-, t_k) \mathbf{P}_k^- \mathbf{H}^T(\hat{\mathbf{x}}_k^-, t_k) + \mathbf{R}_k \right]^{-1}$$

2. Update mean and covariance

$$\begin{aligned}\hat{\mathbf{x}}_k^+ &= \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k^-, t_k)) \\ \mathbf{P}_k^+ &= [\mathbf{I} - \mathbf{K}_k \mathbf{H}(\hat{\mathbf{x}}_k^-, t_k)] \mathbf{P}_k^- [\mathbf{I} - \mathbf{K}_k \mathbf{H}(\hat{\mathbf{x}}_k^-, t_k)]^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T\end{aligned}$$



### 3.6.2 UKF Algorithm

#### Algorithm 2. *UKF*

---

*System and Measurement Model*

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), t) \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k, t_k) + \mathbf{v}_k\end{aligned}$$

*Initialization*

$$\begin{aligned}\hat{\mathbf{x}}_0 &= \mathbb{E} \{ \mathbf{x}(t_0) \} \\ \mathbf{P}_0 &= \mathbb{E} \{ (\mathbf{x}(t_0) - \hat{\mathbf{x}}_0)(\mathbf{x}(t_0) - \hat{\mathbf{x}}_0)^T \}\end{aligned}$$

*Propagation,  $t \in [t_{k-1}, t_k]$*

1. *Determine sigma-points*

$$\begin{aligned}\mathbf{P}_{k-1}^+ &= \mathbf{S}_{k-1} \mathbf{S}_{k-1}^T \\ \mathbf{S}_{k-1} &= [\mathbf{s}_{1,k-1} \quad \dots \quad \mathbf{s}_{n,k-1}] \\ \boldsymbol{\chi}_{i,k-1} &= \hat{\mathbf{x}}_{k-1}^+ + \sqrt{n} \mathbf{s}_{i,k-1} \\ \boldsymbol{\chi}_{i+n,k-1} &= \hat{\mathbf{x}}_{k-1}^+ - \sqrt{n} \mathbf{s}_{i,k-1}\end{aligned}$$

2. *Propagate sigma-points through the dynamics*

$$\dot{\boldsymbol{\chi}}_i(t) = \mathbf{f}(\boldsymbol{\chi}_i(t), t), \quad \boldsymbol{\chi}_i(t_{k-1}) = \boldsymbol{\chi}_{i,k-1}, \quad \boldsymbol{\chi}_{i,k} = \boldsymbol{\chi}_i(t_k)$$

3. Calculate propagated mean and covariance

$$\hat{\mathbf{x}}_k^- = \sum_{i=1}^K w_i \mathbf{x}_{i,k}$$

$$\mathbf{P}_k^- = \sum_{i=1}^K w_i (\mathbf{x}_{i,k} - \hat{\mathbf{x}}_k^-)(\mathbf{x}_{i,k} - \hat{\mathbf{x}}_k^-)^T$$

Update,  $t_k \ \forall \ k = 1, 2, \dots$

1. Compute measurement-transformed sigma-points

$$\mathbf{y}_{i,k} = \mathbf{h}(\mathbf{x}_{i,k}, t_k)$$

2. Compute estimated measurement, measurement covariance, and cross-covariance

$$\hat{\mathbf{y}}_k^- = \sum_{i=1}^K w_i \mathbf{y}_{i,k}$$

$$\mathbf{P}_y = \sum_{i=1}^K w_i (\mathbf{y}_{i,k} - \hat{\mathbf{y}}_k^-)(\mathbf{y}_{i,k} - \hat{\mathbf{y}}_k^-)^T + \mathbf{R}_k$$

$$\mathbf{P}_{xy} = \sum_{i=1}^K w_i (\mathbf{x}_{i,k} - \hat{\mathbf{x}}_k^-)(\mathbf{y}_{i,k} - \hat{\mathbf{y}}_k^-)^T$$

3. Compute Kalman gain

$$\mathbf{K}_k = \mathbf{P}_{xy} \mathbf{P}_y^{-1}$$

4. Update mean and covariance

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - \hat{\mathbf{y}}_k^-)$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_y \mathbf{K}_k^T$$

### 3.6.3 GMEKF Algorithm

#### Algorithm 3. *GMEKF*

---

*System and Measurement Model*

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), t) \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k, t_k) + \mathbf{v}_k\end{aligned}$$

*Initialization*

$$p(\mathbf{x}_0) = \sum_{\ell=1}^L \alpha_{\ell,0} p_g(\mathbf{x}_0; \hat{\mathbf{x}}_{\ell,0}, \mathbf{P}_{\ell,0})$$

*Definitions*

$$\begin{aligned}\mathbf{F}(\hat{\mathbf{x}}_{\ell}(t), t) &= \left[ \frac{\partial \mathbf{f}(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)} \right]_{\mathbf{x}(t)=\hat{\mathbf{x}}_{\ell}(t)} \\ \mathbf{H}(\hat{\mathbf{x}}_{\ell,k}^-, t_k) &= \left[ \frac{\partial \mathbf{h}(\mathbf{x}_k, t_k)}{\partial \mathbf{x}_k} \right]_{\mathbf{x}_k=\hat{\mathbf{x}}_{\ell,k}^-}\end{aligned}$$

*Propagation,  $t \in [t_{k-1}, t_k]$*

1. *Propagate mean and state transition matrix through the dynamics*

$$\begin{aligned}\dot{\hat{\mathbf{x}}}_{\ell}(t) &= \mathbf{f}(\hat{\mathbf{x}}_{\ell}(t), t), \quad \hat{\mathbf{x}}_{\ell}(t_{k-1}) = \hat{\mathbf{x}}_{\ell,k-1}^+, \quad \hat{\mathbf{x}}_{\ell,k}^- = \hat{\mathbf{x}}_{\ell}(t_k) \\ \dot{\Phi}_{\ell}(t, t_{k-1}) &= \mathbf{F}(\hat{\mathbf{x}}_{\ell}(t), t) \Phi_{\ell}(t, t_{k-1}), \quad \Phi_{\ell}(t_{k-1}, t_{k-1}) = \mathbf{I}\end{aligned}$$

2. *Calculate propagated covariance*

$$\mathbf{P}_{\ell}(t) = \Phi_{\ell}(t, t_{k-1}) \mathbf{P}_{\ell}(t_{k-1}) \Phi_{\ell}^T(t, t_{k-1}), \quad \mathbf{P}_{\ell}(t_{k-1}) = \mathbf{P}_{\ell,k-1}^+, \quad \mathbf{P}_{\ell,k}^- = \mathbf{P}_{\ell}(t_k)$$

Update,  $t_k \ \forall \ k = 1, 2, \dots$

1. Compute Kalman gain and weight gain

$$\begin{aligned} \mathbf{K}_{\ell,k} &= \mathbf{P}_{\ell,k}^- \mathbf{H}^T(\hat{\mathbf{x}}_{\ell,k}^-, t_k) [\mathbf{H}(\hat{\mathbf{x}}_{\ell,k}^-, t_k) \mathbf{P}_{\ell,k}^- \mathbf{H}^T(\hat{\mathbf{x}}_{\ell,k}^-, t_k) + \mathbf{R}_k]^{-1} \\ \beta_{\ell,k} &= p_g(\mathbf{y}_k; \mathbf{h}(\hat{\mathbf{x}}_{\ell,k}^-, t_k), \mathbf{H}(\hat{\mathbf{x}}_{\ell,k}^-, t_k) \mathbf{P}_{\ell,k}^- \mathbf{H}^T(\hat{\mathbf{x}}_{\ell,k}^-, t_k) + \mathbf{R}_k) \end{aligned}$$

2. Update weight, mean, and covariance

$$\begin{aligned} \alpha_{\ell,k}^+ &= \frac{\beta_{\ell,k} \alpha_{\ell,k}^-}{\sum_{j=1}^L \beta_{j,k} \alpha_{j,k}^-} \\ \hat{\mathbf{x}}_{\ell,k}^+ &= \hat{\mathbf{x}}_{\ell,k}^- + \mathbf{K}_{\ell,k} (\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{\ell,k}^-, t_k)) \\ \mathbf{P}_{\ell,k}^+ &= [\mathbf{I} - \mathbf{K}_{\ell,k} \mathbf{H}(\hat{\mathbf{x}}_{\ell,k}^-, t_k)] \mathbf{P}_{\ell,k}^- [\mathbf{I} - \mathbf{K}_{\ell,k} \mathbf{H}(\hat{\mathbf{x}}_{\ell,k}^-, t_k)]^T + \mathbf{K}_{\ell,k} \mathbf{R}_k \mathbf{K}_{\ell,k}^T \end{aligned}$$


---

### 3.6.4 GMUKF Algorithm

#### Algorithm 4. *GMUKF*

---

*System and Measurement Model*

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), t) \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k, t_k) + \mathbf{v}_k \end{aligned}$$

*Initialization*

$$p(\mathbf{x}_0) = \sum_{\ell=1}^L \alpha_{\ell,0} p_g(\mathbf{x}_0; \hat{\mathbf{x}}_{\ell,0}, \mathbf{P}_{\ell,0})$$

*Propagation,  $t \in [t_{k-1}, t_k]$*

*1. Determine sigma-points*

$$\mathbf{P}_{\ell,k-1}^+ = \mathbf{S}_{\ell,k-1} \mathbf{S}_{\ell,k-1}^T \quad \text{to find} \quad \mathbf{S}_{\ell,k-1}$$

$$\mathbf{S}_{\ell,k-1} = [\mathbf{s}_{\ell,1,k-1} \quad \dots \quad \mathbf{s}_{\ell,n,k-1}]$$

$$\boldsymbol{\chi}_{\ell,i,k-1} = \hat{\mathbf{x}}_{\ell,k-1}^+ + \sqrt{n} \mathbf{s}_{\ell,i,k-1}$$

$$\boldsymbol{\chi}_{\ell,i+n,k-1} = \hat{\mathbf{x}}_{\ell,k-1}^+ - \sqrt{n} \mathbf{s}_{\ell,i,k-1}$$

*2. Propagate sigma-points through the dynamics*

$$\dot{\boldsymbol{\chi}}_{\ell,i}(t) = \mathbf{f}(\boldsymbol{\chi}_{\ell,i}(t), t), \quad \boldsymbol{\chi}_{\ell,i}(t_{k-1}) = \boldsymbol{\chi}_{\ell,i,k-1}, \quad \boldsymbol{\chi}_{\ell,i,k} = \boldsymbol{\chi}_{\ell,i}(t_k)$$

*3. Calculate propagated mean and covariance*

$$\hat{\mathbf{x}}_{\ell,k}^- = \sum_{i=1}^K w_i \boldsymbol{\chi}_{\ell,i,k}$$

$$\mathbf{P}_{\ell,k}^- = \sum_{i=1}^K w_i (\boldsymbol{\chi}_{\ell,i,k} - \hat{\mathbf{x}}_{\ell,k}^-)(\boldsymbol{\chi}_{\ell,i,k} - \hat{\mathbf{x}}_{\ell,k}^-)^T$$

*Update,  $t_k \quad \forall \quad k = 1, 2, \dots$*

*1. Compute measurement-transformed sigma-points*

$$\mathbf{y}_{\ell,i,k} = \mathbf{h}(\boldsymbol{\chi}_{\ell,i,k}, t_k)$$

*2. Compute estimated measurement, measurement covariance, and*

*cross-covariance*

$$\begin{aligned}\hat{\mathbf{y}}_{\ell,k}^- &= \sum_{i=1}^k w_i \mathbf{y}_{\ell,i,k} \\ \mathbf{P}_{\ell,y} &= \sum_{i=1}^k w_i (\mathbf{y}_{\ell,i,k} - \hat{\mathbf{y}}_{\ell,k}^-)(\mathbf{y}_{\ell,i,k} - \hat{\mathbf{y}}_{\ell,k}^-)^T + \mathbf{R}_k \\ \mathbf{P}_{\ell,xy} &= \sum_{i=1}^k w_i (\mathbf{x}_{\ell,i,k} - \hat{\mathbf{x}}_{\ell,k}^-)(\mathbf{y}_{\ell,i,k} - \hat{\mathbf{y}}_{\ell,k}^-)^T\end{aligned}$$

3. *Compute Kalman gain and weight gain*

$$\begin{aligned}\mathbf{K}_{\ell,k} &= \mathbf{P}_{\ell,xy} \mathbf{P}_{\ell,y}^{-1} \\ \beta_{\ell,k} &= p_g(\mathbf{y}_k; \hat{\mathbf{y}}_{\ell,k}^-, \mathbf{P}_{\ell,y})\end{aligned}$$

4. *Update weight, mean, and covariance*

$$\begin{aligned}\alpha_{\ell,k}^+ &= \frac{\beta_{\ell,k} \alpha_{\ell,k}^-}{\sum_{j=1}^L \beta_{j,k} \alpha_{j,k}^-} \\ \hat{\mathbf{x}}_{\ell,k}^+ &= \hat{\mathbf{x}}_{\ell,k}^- + \mathbf{K}_{\ell,k}(\mathbf{y}_k - \hat{\mathbf{y}}_{\ell,k}^-) \\ \mathbf{P}_{\ell,k}^+ &= \mathbf{P}_{\ell,k}^- - \mathbf{K}_{\ell,k} \mathbf{P}_{\ell,y} \mathbf{K}_{\ell,k}^T\end{aligned}$$

### 3.6.5 SGMUKF Algorithm

**Algorithm 5.** *SGMUKF*

*System and Measurement Model*

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), t) \\ \mathbf{y}_k &= \mathbf{h}(\mathbf{x}_k, t_k) + \mathbf{v}_k\end{aligned}$$

*Initialization*

$$p(\mathbf{x}_0) = \sum_{\ell=1}^L \alpha_{\ell,0} p_g(\mathbf{x}_0; \hat{\mathbf{x}}_{\ell,0}, \mathbf{P}_{\ell,0})$$

*Propagation,  $t \in [t_{k-1}, t_k]$*

1. Set  $t_{s-1} = t_{k-1}$ ,  $\alpha_{\ell,s-1} = \alpha_{\ell,k-1}^+$ ,  $\hat{\mathbf{x}}_{\ell,s-1} = \hat{\mathbf{x}}_{\ell,k-1}^+$ , and  $\mathbf{P}_{\ell,s-1} = \mathbf{P}_{\ell,k-1}^+$

(a) *Determine sigma-points at  $t_{s-1}$*

$$\mathbf{P}_{\ell,s-1} = \mathbf{S}_{\ell,s-1} \mathbf{S}_{\ell,s-1}^T \quad \text{to find} \quad \mathbf{S}_{\ell,s-1}$$

$$\mathbf{S}_{\ell,s-1} = [\mathbf{s}_{\ell,1,s-1} \quad \dots \quad \mathbf{s}_{\ell,n,s-1}]$$

$$\boldsymbol{\chi}_{\ell,i,s-1} = \hat{\mathbf{x}}_{\ell,s-1} + \sqrt{n} \mathbf{s}_{\ell,i,s-1}$$

$$\boldsymbol{\chi}_{\ell,i+n,s-1} = \hat{\mathbf{x}}_{\ell,s-1} - \sqrt{n} \mathbf{s}_{\ell,i,s-1}$$

(b) *Propagate sigma-points through the dynamics until nonlinearity detected at time  $t_s$  on  $j^{\text{th}}$  component*

$$\dot{\boldsymbol{\chi}}_{\ell,i}(t) = \mathbf{f}(\boldsymbol{\chi}_{\ell,i}(t), t), \quad \boldsymbol{\chi}_{\ell,i}(t_{s-1}) = \boldsymbol{\chi}_{\ell,i,s-1}, \quad \boldsymbol{\chi}_{\ell,i,s} = \boldsymbol{\chi}_{\ell,i}(t_s)$$

(c) *Calculate propagated mean and covariance for  $j^{\text{th}}$  component*

$$\hat{\mathbf{x}}_{j,s} = \sum_{i=1}^K w_i \boldsymbol{\chi}_{j,i,s}$$

$$\mathbf{P}_{j,s} = \sum_{i=1}^K w_i (\boldsymbol{\chi}_{j,i,s} - \hat{\mathbf{x}}_{j,s})(\boldsymbol{\chi}_{j,i,s} - \hat{\mathbf{x}}_{j,s})^T$$

(d) *Replace weight, mean, and covariance of  $j^{\text{th}}$  component by splitting into  $G$  components*

$$\alpha_{j,s} p_g(\mathbf{x}; \hat{\mathbf{x}}_{j,s}, \mathbf{P}_{j,s}) \approx \sum_{r=1}^G \alpha_{r,s} p_g(\mathbf{x}; \hat{\mathbf{x}}_{r,s}, \mathbf{P}_{r,s})$$

- (e) Return to Step 1 with  $t_{s-1} = t_s$ ,  $\alpha_{\ell,s-1} = \alpha_{\ell,s}$ ,  $\hat{\mathbf{x}}_{\ell,s-1} = \hat{\mathbf{x}}_{\ell,s}$ , and  $\mathbf{P}_{\ell,s-1} = \mathbf{P}_{\ell,s}$  and continue until  $t_s = t_k$

2. Calculate propagated mean and covariance

$$\hat{\mathbf{x}}_{\ell,k}^- = \sum_{i=1}^K w_i \mathbf{x}_{\ell,i,s}$$

$$\mathbf{P}_{\ell,k}^- = \sum_{i=1}^K w_i (\mathbf{x}_{\ell,i,s} - \hat{\mathbf{x}}_{\ell,k}^-)(\mathbf{x}_{\ell,i,s} - \hat{\mathbf{x}}_{\ell,k}^-)^T$$

Update,  $t_k \ \forall \ k = 1, 2, \dots$

1. Compute measurement-transformed sigma-points

$$\mathbf{y}_{\ell,i,k} = \mathbf{h}(\mathbf{x}_{\ell,i,k}, t_k)$$

2. Compute estimated measurement, measurement covariance, and cross-covariance

$$\hat{\mathbf{y}}_{\ell,k}^- = \sum_{i=1}^k w_i \mathbf{y}_{\ell,i,k}$$

$$\mathbf{P}_{\ell,y} = \sum_{i=1}^k w_i (\mathbf{y}_{\ell,i,k} - \hat{\mathbf{y}}_{\ell,k}^-)(\mathbf{y}_{\ell,i,k} - \hat{\mathbf{y}}_{\ell,k}^-)^T + \mathbf{R}_k$$

$$\mathbf{P}_{\ell,xy} = \sum_{i=1}^k w_i (\mathbf{x}_{\ell,i,k} - \hat{\mathbf{x}}_{\ell,k}^-)(\mathbf{y}_{\ell,i,k} - \hat{\mathbf{y}}_{\ell,k}^-)^T$$

3. Compute Kalman gain and weight gain

$$\mathbf{K}_{\ell,k} = \mathbf{P}_{\ell,xy} \mathbf{P}_{\ell,y}^{-1}$$

$$\beta_{\ell,k} = p_g(\mathbf{y}_k; \hat{\mathbf{y}}_{\ell,k}^-, \mathbf{P}_{\ell,y})$$



4. *Update weight, mean, and covariance*

$$\begin{aligned}\alpha_{\ell,k}^+ &= \frac{\beta_{\ell,k} \alpha_{\ell,k}^-}{\sum_{j=1}^L \beta_{j,k} \alpha_{j,k}^-} \\ \hat{\mathbf{x}}_{\ell,k}^+ &= \hat{\mathbf{x}}_{\ell,k}^- + \mathbf{K}_{\ell,k} (\mathbf{y}_k - \hat{\mathbf{y}}_{\ell,k}^-) \\ \mathbf{P}_{\ell,k}^+ &= \mathbf{P}_{\ell,k}^- - \mathbf{K}_{\ell,k} \mathbf{P}_{\ell,y} \mathbf{K}_{\ell,k}^T\end{aligned}$$

---

## Chapter 4

### Dynamics and Measurement Modeling

#### 4.1 Dynamics Modeling

The dynamics of a spacecraft in orbit are governed by the first-order form of Newton's equations for translation as

$$\begin{aligned}\dot{\mathbf{r}}^i &= \mathbf{v}^i \\ \dot{\mathbf{v}}^i &= \sum \mathbf{a}^i ,\end{aligned}$$

where  $\sum \mathbf{a}^i$  represents the summation of all active forces in the inertial frame. Additionally, the rotational dynamics are governed by the first-order form of Euler's equations as

$$\begin{aligned}\dot{\bar{\mathbf{q}}}_i^b &= \frac{1}{2} \bar{\boldsymbol{\omega}}_{b/i}^b \otimes \bar{\mathbf{q}}_i^b \\ \dot{\boldsymbol{\omega}}_{b/i}^b &= \mathbf{J}^{-1} \left( \sum \mathbf{m}^b - \boldsymbol{\omega}_{b/i}^b \times \mathbf{J} \boldsymbol{\omega}_{b/i}^b \right) ,\end{aligned}$$

where  $\bar{\boldsymbol{\omega}}_{b/i}^b$  is the pure quaternion formed from the angular velocity vector  $\boldsymbol{\omega}_{b/i}^b$ . and  $\otimes$  represents the quaternion multiplication operation, defined such that the quaternions are multiplied in the same order as the equivalent rotation matrices would be. Furthermore,  $\mathbf{J}$  is the moment of inertia of the spacecraft and  $\sum \mathbf{m}^b$  represents the summation of all active moments in the body frame. Letting the active accelerations be the central body gravity (denoted by  $\mathbf{a}_g^i$ ), the third-body gravity (denoted by  $\mathbf{a}_{3\text{rd}}^i$ ),

and the solar radiation pressure (SRP) (denoted by  $\mathbf{a}_{srp}^i$ ), and letting the active moments be the SRP (denoted by  $\mathbf{m}_{srp}^b$ ), the translational and rotational equations of motion may be expressed as

$$\dot{\mathbf{r}}^i = \mathbf{v}^i \quad (4.1a)$$

$$\dot{\mathbf{v}}^i = \mathbf{a}_g^i(\mathbf{r}^i) + \mathbf{a}_{3rd}^i(\mathbf{r}^i) + \mathbf{a}_{srp}^i(\mathbf{r}^i, \bar{\mathbf{q}}_i^b) \quad (4.1b)$$

$$\dot{\bar{\mathbf{q}}}_i^b = \frac{1}{2} \bar{\boldsymbol{\omega}}_{b/i}^b \otimes \bar{\mathbf{q}}_i^b \quad (4.1c)$$

$$\dot{\boldsymbol{\omega}}_{b/i}^b = \mathbf{J}^{-1} \left( \mathbf{m}_{srp}^b(\mathbf{r}^i, \bar{\mathbf{q}}_i^b) - \boldsymbol{\omega}_{b/i}^b \times \mathbf{J} \boldsymbol{\omega}_{b/i}^b \right), \quad (4.1d)$$

where the functional dependencies of the acceleration and moment terms have been included for completeness. If the state is defined by  $\mathbf{x} = [(\mathbf{r}^i)^T (\mathbf{v}^i)^T (\bar{\mathbf{q}}_i^b)^T (\boldsymbol{\omega}_{b/i}^b)^T]^T$ , then Eqs. (4.1) represent the nonlinear dynamical system governing the time evolution of the state, i.e. Eqs. (4.1) represent  $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), t)$ . For linearization-based filtering schemes, such as the extended Kalman filter (EKF) and Gaussian mixture extended Kalman filter (GMEKF), the dynamics are evaluated at the current state estimate, which gives the estimated translational and rotational to be

$$\dot{\hat{\mathbf{r}}}^i = \hat{\mathbf{v}}^i$$

$$\dot{\hat{\mathbf{v}}}^i = \mathbf{a}_g^i(\hat{\mathbf{r}}^i) + \mathbf{a}_{3rd}^i(\hat{\mathbf{r}}^i) + \mathbf{a}_{srp}^i(\hat{\mathbf{r}}^i, \hat{\bar{\mathbf{q}}}_i^b)$$

$$\dot{\hat{\bar{\mathbf{q}}}}_i^b = \frac{1}{2} \hat{\bar{\boldsymbol{\omega}}}_{b/i}^b \otimes \hat{\bar{\mathbf{q}}}_i^b$$

$$\dot{\hat{\boldsymbol{\omega}}}_{b/i}^b = \mathbf{J}^{-1} \left( \mathbf{m}_{srp}^b(\hat{\mathbf{r}}^i, \hat{\bar{\mathbf{q}}}_i^b) - \hat{\boldsymbol{\omega}}_{b/i}^b \times \mathbf{J} \hat{\boldsymbol{\omega}}_{b/i}^b \right).$$

Now, define position, velocity, and angular velocity errors as the difference between the truth and the estimate, yielding

$$\begin{aligned}\delta \mathbf{r}^i &= \mathbf{r}^i - \hat{\mathbf{r}}^i \\ \delta \mathbf{v}^i &= \mathbf{v}^i - \hat{\mathbf{v}}^i \\ \delta \boldsymbol{\omega}_{b/i}^b &= \boldsymbol{\omega}_{b/i}^b - \hat{\boldsymbol{\omega}}_{b/i}^b.\end{aligned}$$

The attitude errors are defined in a different manner since the subtraction of two quaternions would yield a non-quaternion object. For this reason, the attitude error is defined in a multiplicative sense, such that

$$\begin{bmatrix} \frac{1}{2} \delta \boldsymbol{\theta}_i^b \\ 1 \end{bmatrix} = \hat{\mathbf{q}}_i^b \otimes (\hat{\mathbf{q}}_i^b)^{-1},$$

where  $\delta \boldsymbol{\theta}_i^b$  is three-parameter representation of the attitude error, and is typically associated with a vector of small angles. With the definitions of the position, velocity, attitude, and angular velocity errors, their linearized temporal derivatives can be calculated, and are found to be

$$\delta \dot{\mathbf{r}}^i = \delta \mathbf{v}^i \tag{4.2a}$$

$$\delta \dot{\mathbf{v}}^i = (\mathbf{A}_g + \mathbf{A}_{3^{\text{rd}}} + \mathbf{A}_{r,srp}) \delta \mathbf{r}^i + \mathbf{A}_{\theta,srp} \delta \boldsymbol{\theta}_i^b \tag{4.2b}$$

$$\delta \dot{\boldsymbol{\theta}}_i^b = -[\hat{\boldsymbol{\omega}}_{b/i}^b \times] \delta \boldsymbol{\theta}_i^b + \delta \boldsymbol{\omega}_{b/i}^b \tag{4.2c}$$

$$\delta \dot{\boldsymbol{\omega}}_{b/i}^b = \mathbf{J}^{-1} \mathbf{M}_{r,srp} \delta \mathbf{r}^i + \mathbf{J}^{-1} \mathbf{M}_{\theta,srp} \delta \boldsymbol{\theta}_i^b + \mathbf{J}^{-1} [[\mathbf{J} \hat{\boldsymbol{\omega}}_{b/i}^b \times] - [\hat{\boldsymbol{\omega}}_{b/i}^b \times] \mathbf{J}] \delta \boldsymbol{\omega}_{b/i}^b \tag{4.2d}$$

where

$$\begin{aligned}
\mathbf{A}_g &= \left[ \frac{\partial \mathbf{a}_g^i(\mathbf{r}^i)}{\partial \mathbf{r}^i} \right]_{\mathbf{r}^i = \hat{\mathbf{r}}^i} \\
\mathbf{A}_{3\text{rd}} &= \left[ \frac{\partial \mathbf{a}_{3\text{rd}}^i(\mathbf{r}^i)}{\partial \mathbf{r}^i} \right]_{\mathbf{r}^i = \hat{\mathbf{r}}^i} \\
\mathbf{A}_{r,srp} &= \left[ \frac{\partial \mathbf{a}_{srp}^i(\mathbf{r}^i, \bar{\mathbf{q}}_i^b)}{\partial \mathbf{r}^i} \right]_{\substack{\mathbf{r}^i = \hat{\mathbf{r}}^i \\ \bar{\mathbf{q}}_i^b = \hat{\bar{\mathbf{q}}}_i^b}} \\
\mathbf{A}_{\theta,srp} &= \left[ \frac{\partial \mathbf{a}_{srp}^i(\mathbf{r}^i, \bar{\mathbf{q}}_i^b)}{\partial \boldsymbol{\theta}_i^b} \right]_{\substack{\mathbf{r}^i = \hat{\mathbf{r}}^i \\ \bar{\mathbf{q}}_i^b = \hat{\bar{\mathbf{q}}}_i^b}} \\
\mathbf{M}_{r,srp} &= \left[ \frac{\partial \mathbf{m}_{srp}^b(\mathbf{r}^i, \bar{\mathbf{q}}_i^b)}{\partial \mathbf{r}^i} \right]_{\substack{\mathbf{r}^i = \hat{\mathbf{r}}^i \\ \bar{\mathbf{q}}_i^b = \hat{\bar{\mathbf{q}}}_i^b}} \\
\mathbf{M}_{\theta,srp} &= \left[ \frac{\partial \mathbf{m}_{srp}^b(\mathbf{r}^i, \bar{\mathbf{q}}_i^b)}{\partial \boldsymbol{\theta}_i^b} \right]_{\substack{\mathbf{r}^i = \hat{\mathbf{r}}^i \\ \bar{\mathbf{q}}_i^b = \hat{\bar{\mathbf{q}}}_i^b}} .
\end{aligned}$$

Eqs. (4.2) are the linearized dynamics of the estimation error, and therefore represent the elements of the Jacobian matrix for the nonlinear system, which is defined to be

$$\mathbf{F}(\hat{\mathbf{x}}(t), t) = \left[ \frac{\partial \mathbf{f}(\mathbf{x}(t), t)}{\partial \mathbf{x}(t)} \right]_{\mathbf{x}(t) = \hat{\mathbf{x}}(t)} ,$$

which, using Eqs. (4.2) yields

$$\mathbf{F}(\hat{\mathbf{x}}(t), t) = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_g + \mathbf{A}_{3\text{rd}} + \mathbf{A}_{r,srp} & \mathbf{0} & \mathbf{A}_{\theta,srp} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -[\hat{\boldsymbol{\omega}}_{b/i}^b \times] & \mathbf{I} \\ \mathbf{J}^{-1} \mathbf{M}_{r,srp} & \mathbf{0} & \mathbf{J}^{-1} \mathbf{M}_{\theta,srp} & \mathbf{J}^{-1} [\mathbf{J} \hat{\boldsymbol{\omega}}_{b/i}^b \times] - [\hat{\boldsymbol{\omega}}_{b/i}^b \times] \mathbf{J} \end{bmatrix} . \quad (4.3)$$

The remainder of this section is devoted to determining relationships for the evaluation of the accelerations, moments, and their required derivatives so that Eqs. (4.1) and Eq. (4.3) can be evaluated.

#### 4.1.1 Central Body Gravitational Acceleration

We present three methods for describing the gravitational acceleration due to the central body: the point mass model, the zonal harmonics model, and the spherical harmonics model. In each of the methods, we give the form of the model for the gravitational potential and then derive the gravitational acceleration vector and Jacobian matrix. For all of the methods, the gravitational potential can be expressed in functional form as

$$U = U(\mathbf{r}^f, \boldsymbol{\theta}), \quad (4.4)$$

where  $\mathbf{r}^f = \mathbf{T}_i^f \mathbf{r}^i$  is the fixed-frame position of the satellite,  $\mathbf{T}_i^f$  is the transformation of the inertial reference frame to the fixed reference frame,  $\mathbf{r}^i$  is the inertial position of the satellite, and  $\boldsymbol{\theta}$  is the collection of the model parameters (e.g. the gravitational parameter of the central body) into a parameter vector. The first expression of interest is that of the gravitational acceleration. By taking the gradient of Eq. (4.4) with respect to the inertial position, it is readily observed that the inertial gravitational acceleration vector is given by

$$\mathbf{a}_g^i = \mathbf{T}_f^i \mathbf{a}_g^f(\mathbf{r}^f, \boldsymbol{\theta}), \quad (4.5)$$

where

$$\mathbf{a}_g^f(\mathbf{r}^f, \boldsymbol{\theta}) = \left[ \frac{\partial U(\mathbf{r}^f, \boldsymbol{\theta})}{\partial \mathbf{r}^f} \right]^T. \quad (4.6)$$

Furthermore, by taking the gradient of Eq. (4.5) with respect to the inertial position, it is found that

$$\mathbf{A}_g = \frac{\partial \mathbf{a}_g^i}{\partial \mathbf{r}^i} = \mathbf{T}_f^i \mathbf{G}(\mathbf{r}^f, \boldsymbol{\theta}) \mathbf{T}_i^f, \quad (4.7)$$

where

$$\mathbf{G}(\mathbf{r}^f, \boldsymbol{\theta}) = \frac{\partial \mathbf{g}(\mathbf{r}^f, \boldsymbol{\theta})}{\partial \mathbf{r}^f}. \quad (4.8)$$

In the following developments, the form of  $\mathbf{a}_g^f(\mathbf{r}^f, \boldsymbol{\theta})$  and  $\mathbf{G}(\mathbf{r}^f, \boldsymbol{\theta})$  will be derived for each of the associated models of the gravitational potential.

#### 4.1.1.1 Point Mass Gravitational Acceleration

In the point mass model of the gravitational field, the potential is given by

$$U = \frac{\mu}{r},$$

where  $\mu$  is the gravitational parameter of the body and  $r = \|\mathbf{r}^i\| = \|\mathbf{r}^f\|$  is the magnitude of the position vector of the satellite with respect to the center of the body. It is then straightforward to show that the gravitational acceleration vector described in Eq. (4.6) is given by

$$\mathbf{a}_g^f(\mathbf{r}^f, \boldsymbol{\theta}) = -\frac{\mu}{r^3} \mathbf{r}^f, \quad (4.9)$$

and that the Jacobian matrix described in Eq. (4.8) is given by

$$\mathbf{G}(\mathbf{r}^f, \boldsymbol{\theta}) = \frac{\mu}{r^5} (3\mathbf{r}^f(\mathbf{r}^f)^T - r^2 \mathbf{I}). \quad (4.10)$$

Typically, in the implementation of a point mass model of the gravitational field, the orientation of the gravitating body is not utilized since the orientation plays no role in the description of the gravitational field (due to the fact that the model assumes all mass concentrated at a single point and therefore independent of orientation). As such, it is straightforward to show that

$$\mathbf{a}_g^i = -\frac{\mu}{r^3} \mathbf{r}^i, \quad (4.11)$$

and that

$$\frac{\partial \mathbf{a}_g^i}{\partial \mathbf{r}^i} = \frac{\mu}{r^5} (3\mathbf{r}^i (\mathbf{r}^i)^T - r^2 \mathbf{I}) , \quad (4.12)$$

both of which are seen to be orientation independent.

#### 4.1.1.2 Zonal Harmonics Gravitational Acceleration

For a gravitational field modeled with zonal harmonics, the gravitational potential is given by<sup>65</sup>

$$U = -\frac{\mu}{r} \sum_{n=0}^{\infty} \left( \frac{a_e}{r} \right)^n P_n(u) J_n , \quad (4.13)$$

where  $\mu$  is the gravitational parameter of the body,  $a_e$  is the reference distance of the body (usually taken to be the equatorial radius),  $r$  is the distance from the center of the body to the satellite,  $u = \sin \phi$ ,  $\phi$  is the spherical latitude of the satellite,  $J_n$  is the  $n^{th}$  zonal harmonic of the body, and  $P_n(u)$  is the Legendre polynomial of degree  $n$ . The Legendre polynomials are defined as

$$P_n(u) = \frac{1}{2^n n!} \frac{d^n}{du^n} (u^2 - 1)^n ,$$

and can be shown to satisfy the recursions<sup>50,65</sup>

$$P_n(u) = \frac{2n-1}{n} u P_{n-1}(u) = \frac{n-1}{n} P_{n-2}(u) \quad (4.14a)$$

$$\frac{dP_{n+1}(u)}{du} = (n+1)P_n(u) + u \frac{dP_n(u)}{du} . \quad (4.14b)$$

Application of the definition of the Legendre polynomials and their recursion relationships allows us to formulate the functional form of the Legendre polynomials, as is shown in Table 4.1 for degrees 0 to 5.



Table 4.1: Legendre Polynomials and their Derivatives

Degree	Legendre Polynomial	Derivative
0	1	0
1	$u$	1
2	$\frac{1}{2}(3u^2 - 1)$	$3u$
3	$\frac{1}{2}(5u^3 - 3u)$	$\frac{3}{2}(5u^2 - 1)$
4	$\frac{1}{8}(35u^4 - 30u^2 + 3)$	$\frac{5}{2}(7u^3 - 3u)$
5	$\frac{1}{8}(63u^5 - 70u^3 + 15u)$	$\frac{5}{8}(63u^4 - 42u^2 + 3)$

In practical applications, the infinite summation is truncated to enable computation. Typically, low degree representations of the zonal harmonics potential are implemented so as to capture the dominant effects due to asphericity of the body without involving overburdening computation. In the sequel, we will restrict our treatment of the zonal harmonics model to a maximum degree of 4, that is we truncate the infinite summation at 4 to develop equations for the gravitational acceleration vector and Jacobian matrix. However, it should be noted that truncation at a higher degree is merely an extension of the given treatment. In the subsequent developments we leave this as an infinite sum with the understanding that the sum is to be truncated for implementation.

Having established the form of the gravitational potential in Eq. (4.13), we

now turn towards developing a relationship for

$$\mathbf{a}_g^f(\mathbf{r}^f, \boldsymbol{\theta}) = \left[ \frac{\partial U(\mathbf{r}^f, \boldsymbol{\theta})}{\partial \mathbf{r}^f} \right]^T. \quad (4.15)$$

Let the fixed-frame position vector be given by  $\mathbf{r}^f = [x \ y \ z]^T$ , which yields the relationship that  $u = z/r$ , and define

$$g_1 = \frac{\partial U(\mathbf{r}^f, \boldsymbol{\theta})}{\partial x}, \quad g_2 = \frac{\partial U(\mathbf{r}^f, \boldsymbol{\theta})}{\partial y}, \quad \text{and} \quad g_3 = \frac{\partial U(\mathbf{r}^f, \boldsymbol{\theta})}{\partial z},$$

such that Eq. (4.15) becomes

$$\mathbf{a}_g^f(\mathbf{r}^f, \boldsymbol{\theta}) = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix}. \quad (4.16)$$

Differentiating the potential in Eq. (4.13) with respect to  $x$ ,  $y$ , and  $z$  then yields

$$g_1 = \frac{\mu x}{r^3} \sum_{n=0}^{\infty} \left( \frac{a_e}{r} \right)^n \left( (n+1)P_n(u) + u \frac{dP_n(u)}{du} \right) J_n \quad (4.17a)$$

$$g_2 = \frac{\mu y}{r^3} \sum_{n=0}^{\infty} \left( \frac{a_e}{r} \right)^n \left( (n+1)P_n(u) + u \frac{dP_n(u)}{du} \right) J_n \quad (4.17b)$$

$$g_3 = \frac{\mu z}{r^3} \sum_{n=0}^{\infty} \left( \frac{a_e}{r} \right)^n \left( (n+1)P_n(u) + u \frac{dP_n(u)}{du} \right) J_n \quad (4.17c)$$

$$- \frac{\mu}{r^2} \sum_{n=0}^{\infty} \left( \frac{a_e}{r} \right)^n \frac{dP_n(u)}{du} J_n.$$

Utilizing the recursion relationship in Eq. (4.14b), Eqs. (4.17) may be rewritten more compactly as

$$g_1 = \frac{\mu x}{r^3} \sum_{n=0}^{\infty} \left( \frac{a_e}{r} \right)^n \frac{dP_{n+1}(u)}{du} J_n \quad (4.18a)$$

$$g_2 = \frac{\mu y}{r^3} \sum_{n=0}^{\infty} \left( \frac{a_e}{r} \right)^n \frac{dP_{n+1}(u)}{du} J_n \quad (4.18b)$$

$$g_3 = \frac{\mu z}{r^3} \sum_{n=0}^{\infty} \left( \frac{a_e}{r} \right)^n \frac{dP_{n+1}(u)}{du} J_n - \frac{\mu}{r^2} \sum_{n=0}^{\infty} \left( \frac{a_e}{r} \right)^n \frac{dP_n(u)}{du} J_n. \quad (4.18c)$$

Substituting for the Legendre polynomial derivatives from Table 4.1 into Eqs. (4.18), noting that for all gravitational fields  $J_0 = -1$ , and that  $J_1 = 0$  provided that the center of mass coincides with the origin of the coordinate system, it can be shown that the acceleration vector in Eq. (4.16) is given by

$$\mathbf{a}_g^f(\mathbf{r}^f, \boldsymbol{\theta}) = -\frac{\mu}{r^3} \mathbf{r}^f + \sum_{n=2}^{\infty} \frac{\mu a_e^n J_n}{r^{2n+3}} \mathbf{r}_{J_n}, \quad (4.19)$$

where, for  $n = 2$ ,  $n = 3$ , and  $n = 4$ , we have

$$\begin{aligned} \mathbf{r}_{J_2} &= \frac{3}{2} \begin{bmatrix} 5xz^2 - xr^2 \\ 5yz^2 - yr^2 \\ 5z^3 - 3zr^2 \end{bmatrix}, & \mathbf{r}_{J_3} &= \frac{1}{2} \begin{bmatrix} 35xz^3 - 15x zr^2 \\ 35yz^3 - 15y zr^2 \\ 35z^4 - 30z^2 r^2 + 3r^4 \end{bmatrix} \quad \text{and} \\ \mathbf{r}_{J_4} &= \frac{5}{8} \begin{bmatrix} 63xz^4 - 42xz^2 r^2 + 3xr^4 \\ 63yz^4 - 42yz^2 r^2 + 3yr^4 \\ 63z^5 - 70z^3 r^2 + 15zr^4 \end{bmatrix}. \end{aligned}$$

Similar to the acceleration vector, the gravity Jacobian matrix may be found as

$$\mathbf{G}(\mathbf{r}^f, \boldsymbol{\theta}) = \frac{\mu}{r^5} (3\mathbf{r}^f (\mathbf{r}^f)^T - r^2 \mathbf{I}) - \sum_{n=2}^{\infty} \frac{\mu a_e^n J_n}{r^{2n+5}} ((2n+3) \mathbf{r}_{J_n} (\mathbf{r}^f)^T - r^2 \mathbf{G}_{J_n}), \quad (4.20)$$

where, for  $n = 2$ ,  $n = 3$ , and  $n = 4$ , it can be shown that

$$\begin{aligned} \mathbf{G}_{J_2} &= \frac{3}{2} \begin{bmatrix} 5z^2 - 2x^2 - r^2 & -2xy & 8xz \\ -2xy & 5z^2 - 2y^2 - r^2 & 8yz \\ -6xz & -6yz & 9z^2 - 3r^2 \end{bmatrix} \\ \mathbf{G}_{J_3} &= \frac{1}{2} \begin{bmatrix} 35z^3 - 30x^2 z - 15zr^2 & -30xyz & 75xz^2 - 15xr^2 \\ -30xyz & 35z^3 - 30y^2 z - 15zr^2 & 75yz^2 - 15yr^2 \\ -60xz^2 + 12xr^2 & -60yz^2 + 12yr^2 & 80z^3 - 48zr^2 \end{bmatrix} \\ \mathbf{G}_{J_4} &= \frac{5}{8} \begin{bmatrix} 63z^4 - 84x^2 z^2 - 42z^2 r^2 + 12x^2 r^2 + 3r^4 & -84xyz^2 + 12xyr^2 & -140xz^3 + 60x zr^2 & -84xyz^2 + 12xyr^2 \\ -84xyz^2 + 12xyr^2 & 63z^4 - 84y^2 z^2 - 42z^2 r^2 + 12y^2 r^2 + 3r^4 & -140yz^3 + 60y zr^2 & -84xyz^2 + 12xyr^2 \\ -140xz^3 + 60x zr^2 & -140yz^3 + 60y zr^2 & 168xz^3 - 72x zr^2 & 168yz^3 - 72y zr^2 \\ -84xyz^2 + 12xyr^2 & -84xyz^2 + 12xyr^2 & 168xz^3 - 72x zr^2 & 168yz^3 - 72y zr^2 \\ -140xz^3 + 60x zr^2 & -140yz^3 + 60y zr^2 & 175z^4 - 150z^2 r^2 + 15r^4 & 175z^4 - 150z^2 r^2 + 15r^4 \end{bmatrix}. \end{aligned}$$

Therefore, given the fixed-frame position of the satellite, the determination of the gravitational acceleration vector is accomplished via Eq. (4.19) and the gravity Jacobian via Eq. (4.20).

**Numerical Considerations** The appearance of  $r^{2n+3}$  in the denominator of  $\mathbf{a}_g^f(\mathbf{r}^f, \boldsymbol{\theta})$  in Eq. (4.19) and  $r^{2n+5}$  in the denominator of  $\mathbf{G}(\mathbf{r}^f, \boldsymbol{\theta})$  in Eq. (4.20) can potentially present numerical issues when  $r$  is large. As such, it is desirable to reformulate Eqs. (4.19) and (4.20) to avoid this situation. Let us define  $s = x/r$ ,  $t = y/r$ , and recall that  $u = z/r$ . The gravity vector can be written as

$$\mathbf{a}_g^f(\mathbf{r}^f, \boldsymbol{\theta}) = -\frac{\mu}{r^2} \mathbf{u}^f + \sum_{n=2}^{\infty} \frac{\mu J_n}{r^2} \left(\frac{a_e}{r}\right)^n \mathbf{u}_{J_n}, \quad (4.22)$$

where

$$\mathbf{u}^f = \begin{bmatrix} s \\ t \\ u \end{bmatrix},$$

and, for  $n = 2$ ,  $n = 3$ , and  $n = 4$ , we have

$$\begin{aligned} \mathbf{u}_{J_2} &= \frac{3}{2} \begin{bmatrix} 5su^2 - s \\ 5tu^2 - t \\ 5u^3 - 3u \end{bmatrix}, & \mathbf{u}_{J_3} &= \frac{1}{2} \begin{bmatrix} 35su^3 - 15su \\ 35tu^3 - 15tu \\ 35u^4 - 30u^2 + 3 \end{bmatrix} & \text{and} \\ \mathbf{u}_{J_4} &= \frac{5}{8} \begin{bmatrix} 63su^4 - 42su^2 + 3s \\ 63tu^4 - 42tu^2 + 3t \\ 63u^5 - 70u^3 + 15u \end{bmatrix}. \end{aligned}$$

Similarly, the gravity Jacobian matrix of Eq. (4.20) may be rewritten as

$$\mathbf{G}(\mathbf{r}^f, \boldsymbol{\theta}) = \frac{\mu}{r^3} (3\mathbf{u}^f(\mathbf{u}^f)^T - \mathbf{I}) - \sum_{n=2}^{\infty} \frac{\mu J_n}{r^3} \left(\frac{a_e}{r}\right)^n ((2n+3)\mathbf{u}_{J_n}(\mathbf{u}^f)^T - \mathbf{U}_{J_n}), \quad (4.23)$$

where, for  $n = 2$ ,  $n = 3$ , and  $n = 4$ , it can be shown that

$$\begin{aligned}
U_{J_2} &= \frac{3}{2} \left[ \begin{array}{c|c|c} 5u^2 - 2s^2 - 1 & -2st & 8su \\ -2st & 5u^2 - 2t^2 - 1 & 8tu \\ -6su & -6tu & 9u^2 - 3 \end{array} \right] \\
U_{J_3} &= \frac{1}{2} \left[ \begin{array}{c|c|c} 35u^3 - 30s^2u - 15u & -30stu & 75su^2 - 15s \\ -30stu & 35u^3 - 30t^2u - 15u & 75tu^2 - 15t \\ -60su^2 + 12s & -60tu^2 + 12t & 80u^3 - 48u \end{array} \right] \\
U_{J_4} &= \frac{5}{8} \left[ \begin{array}{c|c|c} 63u^4 - 84s^2u^2 - 42u^2 + 12s^2 + 3 & -84stu^2 + 12st & 168su^3 - 72su \\ -84stu^2 + 12st & 63u^4 - 84t^2u^2 - 42u^2 + 12t^2 + 3 & 168tu^3 - 72tu \\ -140su^3 + 60su & -140tu^3 + 60tu & 175u^4 - 150u^2 + 15 \end{array} \right] .
\end{aligned}$$

Thus, to avoid the potential numerical difficulties associated with computation of the gravitational acceleration vector of Eq. (4.19) and the gravity Jacobian of Eq. (4.20), it is recommended to use Eqs. (4.22) and (4.23) instead.

#### 4.1.1.3 Spherical Harmonics Gravitational Acceleration

The form of the spherical harmonics model of the gravitational potential used is that given by Pines.<sup>50</sup> It is known as the uniform representation of the gravitational potential because it serves to remove nonuniform behavior (singularities) from the gravitational acceleration. The uniform representation of the gravitational potential is given by

$$U = \frac{\mu}{r} \sum_{n=0}^{\infty} \sum_{m=0}^n \left( \frac{a_e}{r} \right)^n \bar{A}_{n,m}(u) [\bar{C}_{n,m}r_m(s, t) + \bar{S}_{n,m}i_m(s, t)] , \quad (4.25)$$

where  $\mu$  is the gravitational parameter,  $a_e$  is the reference radius (usually taken as the equatorial radius), and  $\bar{C}_{n,m}$  and  $\bar{S}_{n,m}$  are the normalized spherical harmonics mass

coefficients of the gravitating body. Furthermore,  $r$  is the magnitude of the position vector from the center of mass of the gravitating body to the spacecraft, and  $s$ ,  $t$ , and  $u$  make up the directions of the unit vector pointing to the spacecraft from the center of the body, such that the position unit-vector (expressed in planet-fixed coordinates) is given by

$$\frac{\mathbf{r}^f}{r} = \begin{bmatrix} s \\ t \\ u \end{bmatrix} = \begin{bmatrix} \cos \phi \cos \lambda \\ \cos \phi \sin \lambda \\ \sin \phi \end{bmatrix},$$

where  $\phi$  and  $\lambda$  are the body-centric spherical latitude and longitude, respectively.

$\bar{A}_{n,m}(u)$  is the set of normalized derived Legendre polynomials given by

$$\bar{A}_{n,m}(u) = N_{n,m} A_{n,m}(u) \quad \text{where} \quad A_{n,m}(u) = \frac{1}{2^n n!} \frac{d^{n+m}}{du^{n+m}} (u^2 - 1)^n. \quad (4.26)$$

Here,  $N_{n,m}$  is a normalizing factor which serves to aid in the numerical computation of the spherical harmonics expansion, and is given by

$$N_{n,m} = \left[ \frac{(n-m)! (2n+1) (2 - \delta_{0,m})}{(n+m)!} \right]^{1/2}, \quad \delta_{0,m} = \begin{cases} 1 & , \quad m = 0 \\ 0 & , \quad m > 0 \end{cases}.$$

Finally, the terms  $r_m(s, t)$  and  $i_m(s, t)$  are

$$r_m(s, t) = \text{Re} \{ (s + jt)^m \} \quad \text{and} \quad i_m(s, t) = \text{Im} \{ (s + jt)^m \}, \quad (4.27)$$

where  $\text{Re} \{ \cdot \}$  and  $\text{Im} \{ \cdot \}$  indicate the real and imaginary parts of the input complex-valued number and  $j = \sqrt{-1}$  is the imaginary number. In practical implementations, the infinite sum in Eq. (4.25) is replaced by a finite sum. In subsequent developments we leave this as an infinite sum with the understanding that the sum will be truncated.

**Recursion Relationships** In order for the uniform representation of the gravitational potential to be utilized via computational means, it is necessary to formulate recursion relationships for quantities such as  $\bar{A}_{n,m}(u)$ ,  $r_m(s, t)$ , and  $i_m(s, t)$ . These recursions then allow for faster, more reliable computation of the desired parameters for use in simulation.

**Recursions for  $\bar{A}_{n,m}(u)$**  A more detailed development of the recursion formulas for the non-normalized derived Legendre polynomials is given by Pines<sup>50</sup> and a development of the recursion formulas for the normalized derived Legendre polynomials is given by Lundberg.<sup>39</sup> We can think of the terms  $\bar{A}_{n,m}(u)$  as the elements of a lower-triangular matrix. It is a lower-triangular matrix because all elements which would lie along the diagonal do not involve the parameter  $u$  and hence all elements to the right of diagonal will be zero as seen by the definition of the derived Legendre polynomial. This helps in establishing recursions as “diagonal,” “off-diagonal,” or “column.” Thus  $\bar{A}_{0,0}$  would be the upper leftmost element, increasing  $n$  would increase the row index, and increasing  $m$  would increase the column index. A numerically stable recursion for a column (fixed  $m$  and varying  $n$ ) is given by<sup>39</sup>

$$\begin{aligned} \bar{A}_{n,m}(u) = & \left[ \frac{(2n+1)(2n-1)}{(n+m)(n-m)} \right]^{1/2} u \bar{A}_{n-1,m}(u) \\ & - \left[ \frac{(2n+1)(n+m-1)(n-m-1)}{(2n-3)(n+m)(n-m)} \right]^{1/2} \bar{A}_{n-2,m}(u). \end{aligned} \quad (4.28)$$

Note that this recursion requires the terms  $\bar{A}_{n-1,m}(u)$  and  $\bar{A}_{n-2,m}(u)$  in order to calculate the term  $\bar{A}_{n,m}(u)$ . This means that the two previous elements of the column must be present in order to calculate the current element, such that if given the

diagonal element and the element immediately below it, one entire column of the “matrix” may be determined. Assuming that the diagonal element is known, it can be shown that the element immediately below the diagonal element is given by

$$\bar{A}_{n+1,n}(u) = [(2n+3)]^{1/2} u \bar{A}_{n,n}(u). \quad (4.29)$$

Therefore, if the diagonal of the matrix can be populated then the first off-diagonal can be populated and the above column recursion can be utilized to complete the matrix one column at a time. It can be shown that the diagonal elements of the matrix are determined via the recursion

$$\bar{A}_{n,n}(u) = \left[ \mathcal{S}_n \left( 1 + \frac{1}{2n} \right) \right]^{1/2} \bar{A}_{n-1,n-1}(u) \quad , \quad \mathcal{S}_n = \begin{cases} 2 & , \quad n = 1 \\ 1 & , \quad n > 1 \end{cases} \quad , \quad (4.30)$$

which is initialized with  $\bar{A}_{0,0}(u) = 1$ . Given the value of  $\bar{A}_{0,0}(u)$ , the diagonal terms may be populated using Eq. (4.30), the first off-diagonal terms may be populated using Eq. (4.29) and the columns may be populated one at a time using Eq. (4.28), and therefore the entire set of the normalized derived Legendre polynomials can be obtained for a given value of  $u$ .

**Recursions for  $r_m(s, t)$  and  $i_m(s, t)$**  From the definitions of  $r_m(s, t)$  and  $i_m(s, t)$  given in Eq. (4.27) and manipulation to relate the  $m^{\text{th}}$  terms to the previous terms, it can be shown that  $r_m(s, t)$  and  $i_m(s, t)$  satisfy the recursions

$$r_m(s, t) = sr_{m-1}(s, t) - ti_{m-1}(s, t) \quad \text{and} \quad i_m(s, t) = si_{m-1}(s, t) + tr_{m-1}(s, t),$$

which are initialized via

$$r_0(s, t) = 1 \quad \text{and} \quad i_0(s, t) = 0.$$



**Derivative Relationships** Before computing the actual derivatives of the potential, it is convenient to establish relationships on the derivatives of the terms  $\bar{A}_{n,m}(u)$ ,  $r_m(s, t)$ , and  $i_m(s, t)$ . These relationships will then be used to establish more general derivatives in the subsequent developments.

**Derivatives of  $\bar{A}_{n,m}(u)$**  The set of normalized derived Legendre polynomials is functionally dependent on the parameter  $u$  alone; therefore, the only derivative which will be required is the derivative of the normalized polynomials with respect to the parameter  $u$ . From the definition of the derived Legendre polynomials in Eq. (4.26), it is seen that

$$\frac{\partial}{\partial u} \{A_{n,m}(u)\} = A_{n,m+1}(u).$$

Therefore, utilizing the normalization factor to find the derivative of the normalized derived Legendre polynomials yields

$$\frac{\partial}{\partial u} \{\bar{A}_{n,m}(u)\} = \frac{\partial}{\partial u} \{N_{n,m}A_{n,m}(u)\} = N_{n,m}A_{n,m+1}(u) = \frac{N_{n,m}}{N_{n,m+1}}\bar{A}_{n,m+1}(u).$$

Define a parameter  $\lambda_{n,m}$  to be the ratio of the  $N_{n,m}$  normalization factor to the  $N_{n,m+1}$  normalization factor. Thus,

$$\lambda_{n,m} = \frac{N_{n,m}}{N_{n,m+1}} = [\mathcal{S}_m(n-m)(n+m+1)]^{1/2}, \quad \mathcal{S}_m = \begin{cases} \frac{1}{2} & , \quad m = 0 \\ 1 & , \quad m > 0 \end{cases},$$

and the derivative may be written as

$$\frac{\partial}{\partial u} \{\bar{A}_{n,m}(u)\} = \lambda_{n,m}\bar{A}_{n,m+1}(u). \quad (4.31)$$

**Derivatives of  $r_m(s, t)$  and  $i_m(s, t)$**  The terms  $r_m(s, t)$  and  $i_m(s, t)$  depend functionally only on the parameters  $s$  and  $t$ , and so each terms derivative with respect to the parameters  $s$  and  $t$  must be obtained. From the definition of  $r_m(s, t)$  in Eq. (4.27), it is seen that

$$\frac{\partial r_m(s, t)}{\partial s} = \frac{\partial}{\partial s} \{ \text{Re} \{ (s + jt)^m \} \} = \text{Re} \{ m (s + jt)^{m-1} \} = m r_{m-1}(s, t). \quad (4.32)$$

Similarly, the remaining derivative relationships can be found as

$$\frac{\partial r_m(s, t)}{\partial t} = -m i_{m-1}(s, t), \quad \frac{\partial i_m(s, t)}{\partial s} = m i_{m-1}(s, t), \quad (4.33a)$$

$$\text{and} \quad \frac{\partial i_m(s, t)}{\partial t} = m r_{m-1}(s, t). \quad (4.33b)$$

**The Gravitational Acceleration Vector** Following the process of Pines,<sup>50</sup> it can be shown that the gravitational acceleration vector of Eq. (4.6) is given by

$$\mathbf{g}(\mathbf{r}^f, \boldsymbol{\theta}) = \begin{bmatrix} g_1 + s g_4 \\ g_2 + t g_4 \\ g_3 + u g_4 \end{bmatrix}. \quad (4.34)$$

Define a set of combined mass coefficients as

$$\begin{aligned} \bar{D}_{n,m}(s, t) &= \bar{C}_{n,m} r_m(s, t) + \bar{S}_{n,m} i_m(s, t) \\ \bar{E}_{n,m}(s, t) &= \bar{C}_{n,m} r_{m-1}(s, t) + \bar{S}_{n,m} i_{m-1}(s, t) \\ \bar{F}_{n,m}(s, t) &= \bar{S}_{n,m} r_{m-1}(s, t) - \bar{C}_{n,m} i_{m-1}(s, t) \\ \bar{G}_{n,m}(s, t) &= \bar{C}_{n,m} r_{m-2}(s, t) + \bar{S}_{n,m} i_{m-2}(s, t) \\ \bar{H}_{n,m}(s, t) &= \bar{S}_{n,m} r_{m-2}(s, t) - \bar{C}_{n,m} i_{m-2}(s, t). \end{aligned}$$

Then, making use of the derivative relationships described by Eqs. (4.31)–(4.33), it can be shown that the gravity coefficients are

$$g_1 = \frac{\mu}{r^2} \sum_{n=0}^{\infty} \sum_{m=0}^n \left(\frac{a_e}{r}\right)^n m \bar{A}_{n,m}(u) \bar{E}_{n,m}(s, t) \quad (4.35a)$$

$$g_2 = \frac{\mu}{r^2} \sum_{n=0}^{\infty} \sum_{m=0}^n \left(\frac{a_e}{r}\right)^n m \bar{A}_{n,m}(u) \bar{F}_{n,m}(s, t) \quad (4.35b)$$

$$g_3 = \frac{\mu}{r^2} \sum_{n=0}^{\infty} \sum_{m=0}^n \left(\frac{a_e}{r}\right)^n \lambda_{n,m} \bar{A}_{n,m}(u) \bar{D}_{n,m}(s, t) \quad (4.35c)$$

$$-g_4 = \frac{\mu}{r^2} \sum_{n=0}^{\infty} \sum_{m=0}^n \left(\frac{a_e}{r}\right)^n [(n+m+1) \bar{A}_{n,m}(u) + \lambda_{n,m} u \bar{A}_{n,m+1}(u)] \bar{D}_{n,m}(s, t), \quad (4.35d)$$

where we recall that

$$\lambda_{n,m} = [\mathcal{S}_m(n-m)(n+m+1)]^{1/2}, \quad \mathcal{S}_m = \begin{cases} \frac{1}{2} & , \quad m = 0 \\ 1 & , \quad m > 0 \end{cases}.$$

Note that the  $g_1$  and  $g_2$  are the same as shown by Pines<sup>50</sup> due to the fact that the normalization procedure affects only the derivatives of terms involving the parameter  $u$ . Therefore, while  $g_1$  and  $g_2$  remain the same (modulo the difference caused by normalization) the terms  $g_3$  and  $g_4$  are different.

**The Gravitational Jacobian Matrix** Similar to the development of the gravitational acceleration vector, following the method described in Pines,<sup>50</sup> it can be shown that the gravitational Jacobian of Eq. (4.8) is given by

$$\mathbf{G}(\mathbf{r}^f, \boldsymbol{\theta}) = \begin{bmatrix} g_{11} + 2sg_{41} + s^2g_{44} + g_4/r & g_{12} + tg_{41} - sg_{42} + stg_{44} \\ g_{12} + tg_{41} - sg_{42} + stg_{44} & -g_{11} + 2tg_{42} + t^2g_{44} + g_4/r \\ g_{13} + ug_{41} + sg_{43} + sug_{44} & g_{23} + ug_{42} + tg_{43} + tug_{44} \\ & g_{13} + ug_{41} + sg_{43} + sug_{44} \\ & g_{23} + ug_{42} + tg_{43} + tug_{44} \\ & g_{33} + 2ug_{43} + u^2g_{44} + g_4/r \end{bmatrix}. \quad (4.36)$$

Again, making use of the derivative relationships in Eqs. (4.31)–(4.33), it can be shown that

$$\begin{aligned}
g_{11} &= \frac{\mu}{r^3} \sum_{n=0}^{\infty} \sum_{m=0}^n \left( \frac{a_e}{r} \right)^n m(m-1) \bar{A}_{n,m}(u) \bar{G}_{n,m}(s, t) \\
g_{12} &= \frac{\mu}{r^3} \sum_{n=0}^{\infty} \sum_{m=0}^n \left( \frac{a_e}{r} \right)^n m(m-1) \bar{A}_{n,m}(u) \bar{H}_{n,m}(s, t) \\
g_{13} &= \frac{\mu}{r^3} \sum_{n=0}^{\infty} \sum_{m=0}^n \left( \frac{a_e}{r} \right)^n m \lambda_{n,m} \bar{A}_{n,m+1}(u) \bar{E}_{n,m}(s, t) \\
g_{23} &= \frac{\mu}{r^3} \sum_{n=0}^{\infty} \sum_{m=0}^n \left( \frac{a_e}{r} \right)^n m \lambda_{n,m} \bar{A}_{n,m+1}(u) \bar{F}_{n,m}(s, t) \\
g_{33} &= \frac{\mu}{r^3} \sum_{n=0}^{\infty} \sum_{m=0}^n \left( \frac{a_e}{r} \right)^n m \zeta_{n,m} \bar{A}_{n,m+2}(u) \bar{D}_{n,m}(s, t) \\
-g_{41} &= \frac{\mu}{r^3} \sum_{n=0}^{\infty} \sum_{m=0}^n \left( \frac{a_e}{r} \right)^n \left[ m(n+m+1) \bar{A}_{n,m}(u) + m \lambda_{n,m} u \bar{A}_{n,m+1} \right] \bar{E}_{n,m}(s, t) \\
-g_{42} &= \frac{\mu}{r^3} \sum_{n=0}^{\infty} \sum_{m=0}^n \left( \frac{a_e}{r} \right)^n \left[ m(n+m+1) \bar{A}_{n,m}(u) + m \lambda_{n,m} u \bar{A}_{n,m+1} \right] \bar{F}_{n,m}(s, t) \\
-g_{43} &= \frac{\mu}{r^3} \sum_{n=0}^{\infty} \sum_{m=0}^n \left( \frac{a_e}{r} \right)^n \left[ (n+m+1) \lambda_{n,m} \bar{A}_{n,m+1}(u) + \zeta_{n,m} u \bar{A}_{n,m+2} \right] \bar{D}_{n,m}(s, t) \\
g_{44} &= \frac{\mu}{r^3} \sum_{n=0}^{\infty} \sum_{m=0}^n \left( \frac{a_e}{r} \right)^n \left[ (n+m+1)(n+m+3) \bar{A}_{n,m}(u) \right. \\
&\quad \left. + (2n+2m+4) \lambda_{n,m} u \bar{A}_{n,m+1}(u) + \zeta_{n,m} u^2 \bar{A}_{n,m+2}(u) \right] \bar{D}_{n,m}(s, t),
\end{aligned}$$

where

$$\zeta_{n,m} = [\mathcal{S}_m(n-m)(n-m-1)(n+m+1)(n+m+2)]^{1/2}, \quad \mathcal{S}_m = \begin{cases} \frac{1}{2} & , \quad m=0 \\ 1 & , \quad m>0 \end{cases}.$$

#### 4.1.2 Third-Body Gravitational Acceleration

Beyond the effect of the central body gravitational acceleration, a satellite experiences the effects of the gravitational acceleration of the Sun, Moon, and the planets. While less dominant than the central body gravitational acceleration, the third-body effect can produce perturbations to the satellite orbit. The third-body gravitational acceleration is modeled as<sup>65</sup>

$$\mathbf{a}_{3\text{rd}}^i = \sum_{j=1}^k \mu_j \left( \frac{\mathbf{d}_j^i}{d_j^3} - \frac{\mathbf{r}_j^i}{r_j^3} \right), \quad (4.37)$$

where  $\mu_j$  is the gravitational parameter of body  $j$ ,  $\mathbf{d}_j^i = \mathbf{r}_j^i - \mathbf{r}^i$  is the position of body  $j$  with respect to the satellite,  $\mathbf{r}_j^i$  is the position of body  $j$  with respect to the central body,  $\mathbf{r}^i$  is the position of the satellite with respect to the central body, and  $k$  is the number of bodies under consideration. As with the central body acceleration, it is also of interest to develop a relationship for the Jacobian of the third-body gravitational acceleration with respect to the inertial position of the satellite, that is

$$\mathbf{A}_{3\text{rd}} = \frac{\partial \mathbf{a}_{3\text{rd}}^i}{\partial \mathbf{r}^i}.$$

Differentiating Eq. (4.37) with respect to the inertial position of the satellite yields the  $n$ -body Jacobian as

$$\mathbf{A}_{3\text{rd}} = \sum_{j=1}^k \frac{\mu_j}{d_j^5} (3\mathbf{d}_j^i (\mathbf{d}_j^i)^T - d_j^2 \mathbf{I}). \quad (4.38)$$

### 4.1.3 Solar Radiation Pressure Acceleration

The acceleration due to SRP acting on a body with  $n$  flat plates can be expressed as<sup>8</sup>

$$\mathbf{a}_{srp}^i = -s_f f \left( \frac{r_{au}}{r_{s/o}} \right)^2 \sum_{k=1}^n \frac{A_k}{m} \cos \phi_k \left[ (1 - \rho_k) \mathbf{u}_{sun}^i + 2 \left( \frac{1}{3} \delta_k + \cos \phi_k \rho_k \right) \mathbf{u}_{n,k}^i \right], \quad (4.39)$$

where  $s_f$  is the solar flux constant,  $f \in [0, 1]$  is a shadowing factor (see Section 4.1.5) that accounts for solar eclipsing by the Earth,  $r_{au}$  is the distance of one astronomical unit,  $r_{s/o} = \|\mathbf{r}_{sun}^i - \mathbf{r}^i\|$  is the distance of the Sun with respect the satellite,  $m$  is the total satellite mass,  $A_k$  is the area of the  $k^{\text{th}}$  plate,  $\rho_k$  is the specular reflection coefficient of the  $k^{\text{th}}$  plate,  $\delta_k$  is the diffuse reflection coefficient of the  $k^{\text{th}}$  plate,  $\mathbf{u}_{n,k}^i = \mathbf{T}_b^i \mathbf{u}_{n,k}^b$ , where  $\mathbf{u}_{n,k}^b$  is the unit vector normal to the  $k^{\text{th}}$  plate expressed in the satellite body reference frame, and  $\phi_k$  is the angle of incidence of the sunlight with respect to the plate normal, such that

$$\cos \phi_k = \mathbf{u}_{n,k}^i \cdot \mathbf{u}_{sun}^i.$$

Additionally,  $\mathbf{u}_{sun}^i$  is the unit vector from the satellite to the Sun expressed in the inertial reference frame, which is given by

$$\mathbf{u}_{sun}^i = \frac{\mathbf{r}_{sun}^i - \mathbf{r}^i}{\|\mathbf{r}_{sun}^i - \mathbf{r}^i\|},$$

where  $\mathbf{r}_{sun}^i$  is the position of the Sun with respect to the Earth, and  $\mathbf{r}^i$  is the position of the satellite with respect to the Earth. In the sequel it will be useful to rewrite Eq. (4.39) as

$$\mathbf{a}_{srp}^i = -s_f f \left( \frac{r_{au}}{r_{s/o}} \right)^2 \sum_{k=1}^n \frac{A_k}{m} \cos \phi_k \mathbf{u}_{srp,k}^i, \quad (4.40)$$

where

$$\mathbf{u}_{srp,k}^i = (1 - \rho_k) \mathbf{u}_{\text{sun}}^i + 2 \left( \frac{1}{3} \delta_k + \cos \phi_k \rho_k \right) \mathbf{u}_{n,k}^i.$$

Taking the partial derivative of  $\mathbf{a}_{srp}^i$  in Eq. (4.40) with respect to the satellite position,  $\mathbf{r}^i$  yields

$$\begin{aligned} \mathbf{A}_{r,srp} = \frac{\partial \mathbf{a}_{srp}^i}{\partial \mathbf{r}^i} = & -s_f f \sum_{k=1}^n \frac{A_k}{m} \cos \phi_k \mathbf{u}_{srp,k}^i \frac{\partial}{\partial \mathbf{r}^i} \left\{ \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \right\} \\ & - s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \sum_{k=1}^n \frac{A_k}{m} \mathbf{u}_{srp,k}^i \frac{\partial \cos \phi_k}{\partial \mathbf{r}^i} \\ & - s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \sum_{k=1}^n \frac{A_k}{m} \cos \phi_k \frac{\partial \mathbf{u}_{srp,k}^i}{\partial \mathbf{r}^i}, \end{aligned} \quad (4.41)$$

where it is noted that the shadowing factor variation due to position variations is not included. This term is omitted since there should be no variation in the SRP acceleration when the satellite is in either full sunlight or full shadow, which would be contradicted by including a variation of the shadowing function when computing the derivative of the acceleration with respect to position. The three partial derivatives appearing in Eq. (4.41) can readily be determined as

$$\frac{\partial}{\partial \mathbf{r}^i} \left\{ \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \right\} = 2 \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \frac{1}{r_{s/o}} (\mathbf{u}_{\text{sun}}^i)^T \quad (4.42a)$$

$$\frac{\partial \cos \phi_k}{\partial \mathbf{r}^i} = (\mathbf{u}_{n,k}^i)^T \frac{\partial \mathbf{u}_{\text{sun}}^i}{\partial \mathbf{r}^i} \quad (4.42b)$$

$$\frac{\partial \mathbf{u}_{srp,k}^i}{\partial \mathbf{r}^i} = [(1 - \rho_k) \mathbf{I} + 2\rho_k \mathbf{u}_{n,k}^i (\mathbf{u}_{n,k}^i)^T] \frac{\partial \mathbf{u}_{\text{sun}}^i}{\partial \mathbf{r}^i}. \quad (4.42c)$$

Additionally, the derivative of the unit vector from the satellite to the Sun with respect to the position is given by

$$\frac{\partial \mathbf{u}_{\text{sun}}^i}{\partial \mathbf{r}^i} = \frac{1}{r_{s/o}} [-\mathbf{I} + \mathbf{u}_{\text{sun}}^i (\mathbf{u}_{\text{sun}}^i)^T] = \frac{1}{r_{s/o}} [\mathbf{u}_{\text{sun}}^i \times]^2. \quad (4.43)$$

Substituting Eqs. (4.42) and (4.43) into Eq. (4.41) then yields the derivative of the SRP acceleration with respect to the position to be

$$\begin{aligned}\mathbf{A}_{r,srp} = & -2s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \frac{1}{r_{s/o}} \sum_{k=1}^n \frac{A_k}{m} \cos \phi_k \mathbf{u}_{srp,k}^i (\mathbf{u}_{\text{sun}}^i)^T \\ & - s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \frac{1}{r_{s/o}} \sum_{k=1}^n \frac{A_k}{m} \mathbf{u}_{srp,k}^i (\mathbf{u}_{n,k}^i)^T [\mathbf{u}_{\text{sun}}^i \times]^2 \\ & - s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \frac{1}{r_{s/o}} \sum_{k=1}^n \frac{A_k}{m} \cos \phi_k [(1 - \rho_k) \mathbf{I} + 2\rho_k \mathbf{u}_{n,k}^i (\mathbf{u}_{n,k}^i)^T] [\mathbf{u}_{\text{sun}}^i \times]^2.\end{aligned}$$

Taking the partial derivative of  $\mathbf{a}_{srp}^i$  in Eq. (4.40) with respect to the three-dimensional rotation vector,  $\boldsymbol{\theta}_i^b$  yields

$$\begin{aligned}\mathbf{A}_{\theta,srp} = \frac{\partial \mathbf{a}_{srp}^i}{\partial \boldsymbol{\theta}_i^b} = & -s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \sum_{k=1}^n \frac{A_k}{m} \mathbf{u}_{srp,k}^i \frac{\partial \cos \phi_k}{\partial \boldsymbol{\theta}_i^b} \\ & - s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \sum_{k=1}^n \frac{A_k}{m} \cos \phi_k \frac{\partial \mathbf{u}_{srp,k}^i}{\partial \boldsymbol{\theta}_i^b}.\end{aligned}\tag{4.44}$$

The two partial derivatives appearing in Eq. (4.44) are given by

$$\frac{\partial \cos \phi_k}{\partial \boldsymbol{\theta}_i^b} = (\mathbf{u}_{\text{sun}}^i)^T \frac{\partial \mathbf{u}_{n,k}^i}{\partial \boldsymbol{\theta}_i^b}\tag{4.45a}$$

$$\frac{\partial \mathbf{u}_{srp,k}^i}{\partial \boldsymbol{\theta}_i^b} = \left[ 2 \left( \frac{1}{3} \delta_k + \cos \phi_k \rho_k \right) \mathbf{I} + 2\rho_k \mathbf{u}_{n,k}^i (\mathbf{u}_{\text{sun}}^i)^T \right] \frac{\partial \mathbf{u}_{n,k}^i}{\partial \boldsymbol{\theta}_i^b}.\tag{4.45b}$$

Additionally, it can be shown that the derivative of the plate-normal unit vector with respect to the rotation vector is given by

$$\frac{\partial \mathbf{u}_{n,k}^i}{\partial \boldsymbol{\theta}_i^b} = -\mathbf{T}_b^i [\mathbf{u}_{n,k}^b \times].\tag{4.46}$$

Substituting Eqs. (4.45) and (4.46) into Eq. (4.44) then yields the derivative of the



SRP acceleration with respect to the rotation vector to be

$$\begin{aligned} \mathbf{A}_{\theta, srp} = & s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \sum_{k=1}^n \frac{A_k}{m} \mathbf{u}_{srp,k}^i (\mathbf{u}_{\text{sun}}^i)^T \mathbf{T}_b^i [\mathbf{u}_{n,k}^b \times] \\ & + s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \sum_{k=1}^n \frac{A_k}{m} \cos \phi_k \left[ 2 \left( \frac{1}{3} \delta_k + \cos \phi_k \rho_k \right) \mathbf{I} + 2 \rho_k \mathbf{u}_{n,k}^i (\mathbf{u}_{\text{sun}}^i)^T \right] \\ & \times \mathbf{T}_b^i [\mathbf{u}_{n,k}^b \times]. \end{aligned} \quad (4.47)$$

#### 4.1.4 Solar Radiation Pressure Moment

In order to compute the moment induced on the satellite by SRP we consider at the force acting on the satellite due to the SRP, which is given in the inertial frame by multiplying the acceleration in Eq. (4.40) by the total satellite mass, such that

$$\mathbf{f}_{srp}^i = -s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \sum_{k=1}^n A_k \cos \phi_k \mathbf{u}_{srp,k}^i.$$

Since Euler's equations require the moment in the satellite body reference frame we then express the force in the body reference frame as  $\mathbf{f}_{srp}^b = \mathbf{T}_i^b \mathbf{f}_{srp}^i$ , or

$$\mathbf{f}_{srp}^b = -s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \sum_{k=1}^n A_k \cos \phi_k \mathbf{u}_{srp,k}^b,$$

where

$$\mathbf{u}_{srp,k}^b = (1 - \rho_k) \mathbf{u}_{\text{sun}}^b + 2 \left( \frac{1}{3} \delta_k + \cos \phi_k \rho_k \right) \mathbf{u}_{n,k}^b.$$

Here,  $\mathbf{u}_{\text{sun}}^b = \mathbf{T}_i^b \mathbf{u}_{\text{sun}}^i$  and the definition given previously for  $\mathbf{u}_{\text{sun}}^i$  remains valid. Similarly, it is noted that  $\mathbf{u}_{n,k}^b$  does not need to be rotated into the inertial frame as was necessary for the computation of the acceleration. Furthermore, since the angle of incidence of the sunlight with respect to the plate normal is not affected by the

reference frame, we rewrite  $\cos \phi_k$  in terms of the body frame plate-normal unit vector and the body frame unit vector to the Sun from the satellite, yielding

$$\cos \phi_k = \mathbf{u}_{n,k}^b \cdot \mathbf{u}_{\text{sun}}^b.$$

The moment due to SRP is computed as the sum of moments on each plate via

$$\mathbf{m}_{srp}^b = -s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \sum_{k=1}^n A_k \cos \phi_k (\mathbf{r}_{p,k}^b \times \mathbf{u}_{srp,k}^b), \quad (4.48)$$

where  $\mathbf{r}_{p,k}^b$  is the position of the  $k^{\text{th}}$  plate with respect to the satellite center of mass as expressed in the body reference frame.

Taking the partial derivative of  $\mathbf{m}_{srp}^b$  in Eq. (4.48) with respect to the satellite position,  $\mathbf{r}^i$  yields

$$\begin{aligned} \mathbf{M}_{r,srp} = \frac{\partial \mathbf{m}_{srp}^b}{\partial \mathbf{r}^i} = & -s_f f \sum_{k=1}^n A_k \cos \phi_k (\mathbf{r}_{p,k}^b \times \mathbf{u}_{srp,k}^b) \frac{\partial}{\partial \mathbf{r}^i} \left\{ \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \right\} \\ & - s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \sum_{k=1}^n A_k (\mathbf{r}_{p,k}^b \times \mathbf{u}_{srp,k}^b) \frac{\partial \cos \phi_k}{\partial \mathbf{r}^i} \\ & - s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \sum_{k=1}^n A_k \cos \phi_k [\mathbf{r}_{p,k}^b \times] \frac{\partial \mathbf{u}_{srp,k}^b}{\partial \mathbf{r}^i}. \end{aligned} \quad (4.49)$$

As with the acceleration, we omit the variation in the SRP moment due to shadowing variations, that is we do not include the partial derivative of the shadowing function with respect to the satellite position. The three partial derivatives appearing in Eq. (4.49) can readily be determined as

$$\frac{\partial}{\partial \mathbf{r}^i} \left\{ \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \right\} = 2 \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \frac{1}{r_{s/o}} (\mathbf{u}_{\text{sun}}^i)^T \quad (4.50a)$$

$$\frac{\partial \cos \phi_k}{\partial \mathbf{r}^i} = (\mathbf{u}_{n,k}^b)^T \frac{\partial \mathbf{u}_{\text{sun}}^b}{\partial \mathbf{r}^i} \quad (4.50b)$$

$$\frac{\partial \mathbf{u}_{srp,k}^b}{\partial \mathbf{r}^i} = [(1 - \rho_k) \mathbf{I} + 2\rho_k \mathbf{u}_{n,k}^b (\mathbf{u}_{n,k}^b)^T] \frac{\partial \mathbf{u}_{\text{sun}}^b}{\partial \mathbf{r}^i}. \quad (4.50c)$$

Additionally, the derivative of the body-frame unit vector from the satellite to the Sun with respect to the position is given by

$$\frac{\partial \mathbf{u}_{\text{sun}}^b}{\partial \mathbf{r}^i} = \mathbf{T}_i^b \frac{\partial \mathbf{u}_{\text{sun}}^i}{\partial \mathbf{r}^i} = \frac{1}{r_{s/o}} \mathbf{T}_i^b [-\mathbf{I} + \mathbf{u}_{\text{sun}}^i (\mathbf{u}_{\text{sun}}^i)^T] = \frac{1}{r_{s/o}} \mathbf{T}_i^b [\mathbf{u}_{\text{sun}}^i \times]^2. \quad (4.51)$$

Substituting Eqs. (4.50) and (4.51) into Eq. (4.49) then yields the derivative of the SRP moment with respect to the position to be

$$\begin{aligned} \mathbf{M}_{r, \text{srp}} = & -2s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \frac{1}{r_{s/o}} \sum_{k=1}^n A_k \cos \phi_k (\mathbf{r}_{p,k}^b \times \mathbf{u}_{\text{srp},k}^b) (\mathbf{u}_{\text{sun}}^i)^T \\ & - s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \frac{1}{r_{s/o}} \sum_{k=1}^n A_k (\mathbf{r}_{p,k}^b \times \mathbf{u}_{\text{srp},k}^b) (\mathbf{u}_{n,k}^b)^T \mathbf{T}_i^b [\mathbf{u}_{\text{sun}}^i \times]^2 \\ & - s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \frac{1}{r_{s/o}} \sum_{k=1}^n A_k \cos \phi_k [\mathbf{r}_{p,k}^b \times] [(1 - \rho_k) \mathbf{I} + 2\rho_k \mathbf{u}_{n,k}^b (\mathbf{u}_{n,k}^b)^T] \\ & \times \mathbf{T}_i^b [\mathbf{u}_{\text{sun}}^i \times]^2. \end{aligned}$$

Taking the partial derivative of  $\mathbf{m}_{\text{srp}}^b$  in Eq. (4.48) with respect to the three-dimensional rotation vector,  $\boldsymbol{\theta}_i^b$  yields

$$\begin{aligned} \mathbf{M}_{\theta, \text{srp}} = \frac{\partial \mathbf{m}_{\text{srp}}^b}{\partial \boldsymbol{\theta}_i^b} = & -s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \sum_{k=1}^n A_k (\mathbf{r}_{p,k}^b \times \mathbf{u}_{\text{srp},k}^b) \frac{\partial \cos \phi_k}{\partial \boldsymbol{\theta}_i^b} \\ & - s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \sum_{k=1}^n A_k \cos \phi_k [\mathbf{r}_{p,k}^b \times] \frac{\partial \mathbf{u}_{\text{srp},k}^b}{\partial \boldsymbol{\theta}_i^b}. \end{aligned} \quad (4.52)$$

The two partial derivatives appearing in Eq. (4.52) are given by

$$\frac{\partial \cos \phi_k}{\partial \boldsymbol{\theta}_i^b} = (\mathbf{u}_{n,k}^b)^T \frac{\partial \mathbf{u}_{\text{sun}}^b}{\partial \boldsymbol{\theta}_i^b} \quad (4.53a)$$

$$\frac{\partial \mathbf{u}_{\text{srp},k}^b}{\partial \boldsymbol{\theta}_i^b} = [(1 - \rho_k) \mathbf{I} + 2\rho_k \mathbf{u}_{n,k}^b (\mathbf{u}_{n,k}^b)^T] \frac{\partial \mathbf{u}_{\text{sun}}^b}{\partial \boldsymbol{\theta}_i^b}. \quad (4.53b)$$

Additionally, it can be shown that the derivative of the body-frame unit vector to the Sun with respect to the rotation vector is given by

$$\frac{\partial \mathbf{u}_{\text{sun}}^b}{\partial \boldsymbol{\theta}_i^b} = [\mathbf{u}_{\text{sun}}^b \times]. \quad (4.54)$$

Substituting Eqs. (4.53) and (4.54) into Eq. (4.52) then yields the derivative of the SRP moment with respect to the rotation vector to be

$$\begin{aligned} \mathbf{M}_{\theta, \text{srp}} = & -s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \sum_{k=1}^n A_k (\mathbf{r}_{p,k}^b \times \mathbf{u}_{\text{srp},k}^b) (\mathbf{u}_{n,k}^b)^T [\mathbf{u}_{\text{sun}}^b \times] \\ & - s_f f \left( \frac{r_{\text{au}}}{r_{s/o}} \right)^2 \sum_{k=1}^n A_k \cos \phi_k [\mathbf{r}_{p,k}^b \times] [(1 - \rho_k) \mathbf{I} + 2\rho_k \mathbf{u}_{n,k}^b (\mathbf{u}_{n,k}^b)^T] [\mathbf{u}_{\text{sun}}^b \times]. \end{aligned}$$

#### 4.1.5 Models for the Shadow Factor

##### 4.1.5.1 Cylindrical Model

The simplest model for the shadow factor is a cylindrical shadow model. The cylindrical model assumes that the Sun is infinitely far away from the Earth, thus causing the light rays to be completely parallel which yields a cylindrical shadow extending behind the Earth with respect to the Sun, as shown in Figure 4.1. Consider now two vectors: the position of the Sun with respect to the Earth,  $\mathbf{r}_{\text{sun}}^i$ , and the position of the satellite with respect to the Earth,  $\mathbf{r}^i$ . Let the angle between these two vectors be  $\psi$ , such that

$$\cos \psi = \frac{\mathbf{r}^i \cdot \mathbf{r}_{\text{sun}}^i}{\|\mathbf{r}^i\| \|\mathbf{r}_{\text{sun}}^i\|}. \quad (4.55)$$

Then, if  $\cos \psi \geq 0$ , the satellite is on the Sun side of the Earth, meaning that it is illuminated. As such, the shadow factor for the cylindrical model in this case will be

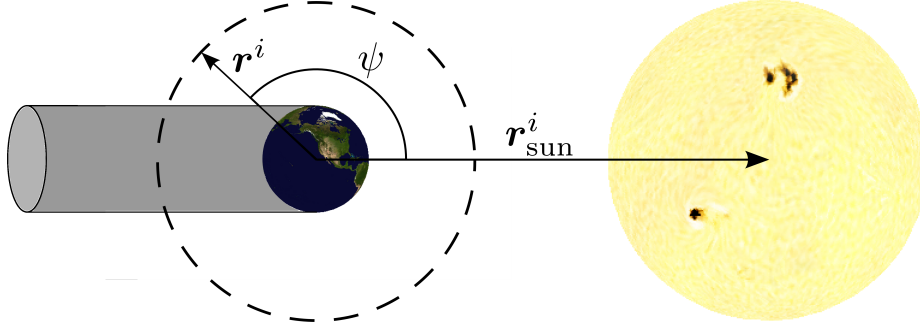


Figure 4.1: Cylindrical Shadow Model

$f_c = 1$ . However, if  $\cos \psi < 0$  then the satellite is on the shadow side of the Earth, but not necessarily in shadow. In this case, if the perpendicular distance of the satellite from the Sun-Earth line is greater than the Earth radius, then the satellite is again illuminated, that is if

$$\|\mathbf{r}^i\|^2(1 - \cos^2 \psi) \geq R_e^2, \quad (4.56)$$

then the shadow factor for the cylindrical model is  $f_c = 1$ . If  $\cos \psi < 0$  and the condition in Eq. (4.56) is not met, then  $f_c = 0$ . The cylindrical shadow factor can therefore be summarized as

$$f_c = \begin{cases} 0 & , \quad \cos \psi < 0 \quad \text{and} \quad \|\mathbf{r}^i\|^2(1 - \cos^2 \psi) < R_e^2 \\ 1 & , \quad \text{otherwise} \end{cases}.$$

#### 4.1.5.2 Conic Model

A level of refinement above the cylindrical model introduces the utilization of cones to model the umbra/penumbra shadowing of the Sun due to the presence of the Earth. This model does not assume that the rays of light emitted by the

Sun are parallel, and is therefore able to more realistically characterize the nature of shadowing. As the Sun moves behind the Earth (i.e. as the satellite moves into the penumbra), the geometry is defined as shown in Figure 4.2.

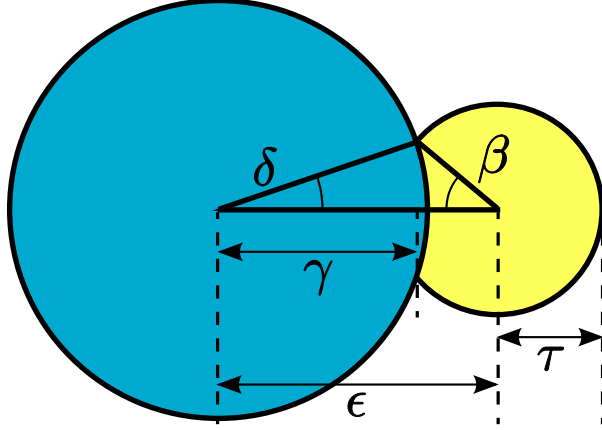


Figure 4.2: Geometry of the Penumbra Phase

Let the radius of the Earth be  $R_e$ , the radius of the Sun be  $R_s$ , the position of the Sun with respect to the Earth be  $\mathbf{r}_{\text{sun}}^i$ , and the position of the satellite with respect to the Earth be  $\mathbf{r}^i$ . It can be shown that  $\gamma$ ,  $\tau$ , and  $\epsilon$  as defined in Figure 4.2 are given by<sup>19</sup>

$$\gamma = \sin^{-1} \frac{R_e}{\|\mathbf{r}^i\|}, \quad \tau = \sin^{-1} \frac{R_s}{\|\mathbf{r}_{\text{sun}}^i - \mathbf{r}^i\|} \quad \text{and} \quad \epsilon = \cos^{-1} \frac{(\mathbf{r}_{\text{sun}}^i - \mathbf{r}^i) \cdot \mathbf{r}^i}{\|\mathbf{r}_{\text{sun}}^i - \mathbf{r}^i\| \|\mathbf{r}^i\|}.$$

Then, we compute  $s$  and  $k$  via

$$s = \frac{1}{2}(\tau + \gamma + \epsilon) \quad \text{and} \quad k = \sqrt{s(s - \tau)(s - \gamma)(s - \epsilon)},$$

such that the angles  $\delta$  and  $\beta$  (shown in Figure 4.2) may be determined as

$$\delta = \tan^{-1} \frac{4k}{\epsilon^2 + \gamma^2 - \tau^2} \quad \text{and} \quad \beta = \tan^{-1} \frac{4k}{\epsilon^2 + \tau^2 - \gamma^2}.$$

Then, the shadow factor for the conic model is given by

$$f_{dc} = 1 - \frac{1}{\pi} \left[ \left( \frac{\gamma}{\tau} \right)^2 \left( \delta - \frac{1}{2} \sin 2\delta \right) + \left( \beta - \frac{1}{2} \sin 2\beta \right) \right] .$$

## 4.2 Measurement Modeling

The optical observation of a satellite from a telescope is given by the right ascension and declination angles of the line-of-sight from the telescope to the satellite. This is a straightforward computation given the position of the satellite and of the telescope; however, key effects must be considered in order to properly determine the line-of-sight, and subsequently, the right ascension and declination angles. The effects that must be considered are the light time correction, stellar aberration correction, satellite lighting condition, telescope lighting condition, and field of view condition. In the following developments, we discuss each of these effects individually, culminating in a combination of the effects to comprise a model for the right ascension and declination measurements made of a satellite by an optical telescope. After describing the model, we develop the first-order derivative relationships associated with the measurement model.

### 4.2.1 Light Time Correction

Due to the finite velocity of light, the time at which photons are reflected off of a satellite differs from that at which they are received at a telescope tracking station. That is, if light reflected from a satellite reaches the telescope at time  $t_k$ , then that light actually reflected off of the satellite at time  $t_k - \lambda$ , where  $\lambda$  is the one-way light time. This effect should be accounted for so that the actual position of the satellite

may be used in the computation of the observed line-of-sight. An in-depth discussion of the light time correction including relativistic effects may be found in Refs. 46,47. For the purposes here, we will ignore the relativistic corrections and focus only on a purely Newtonian approach. Therefore, the light time  $\lambda$  is the unique value which satisfies<sup>5</sup>

$$\lambda = \frac{1}{c} \left\| [\mathbf{r}^i(t_k - \lambda) - \mathbf{r}_{\text{ssb}}^i(t_k - \lambda)] - [\mathbf{r}_{\text{stn}}^i(t_k) - \mathbf{r}_{\text{ssb}}^i(t_k)] \right\| , \quad (4.57)$$

where  $c$  is the speed of light,  $\mathbf{r}^i$  is the position of the satellite with respect to the Earth,  $\mathbf{r}_{\text{ssb}}^i$  is the position of the solar system barycenter with respect to the Earth, and  $\mathbf{r}_{\text{stn}}^i$  is the position of the telescope with respect to the Earth. Each of the position vectors is time-dependent as indicated by the argument following each of the terms in Eq. (4.57). It is assumed that  $\mathbf{r}_{\text{ssb}}^i$  is available in an ephemeris file, such that it can be evaluated at time  $t_k$  as well as at time  $t_k - \lambda$ . To arbitrary precision  $\lambda$  may be computed as<sup>5</sup>

$$\lambda_i = \frac{1}{c} \left\| [\mathbf{r}^i(t_k - \lambda_{i-1}) - \mathbf{r}_{\text{ssb}}^i(t_k - \lambda_{i-1})] - [\mathbf{r}_{\text{stn}}^i(t_k) - \mathbf{r}_{\text{ssb}}^i(t_k)] \right\| , \quad (4.58)$$

for  $i = 1, 2, 3, \dots$ , which is initialized with  $\lambda_0 = 0$ . Typically,  $\lambda = \lambda_1$  is sufficient since, so long as the satellite and telescope are less than 50 astronomical units apart, the error in the computation of the light time via  $\lambda_1$  is less than one millisecond. That is,  $\lambda$  is well approximated as

$$\lambda \approx \frac{1}{c} \left\| \mathbf{r}^i(t_k) - \mathbf{r}_{\text{stn}}^i(t_k) \right\| . \quad (4.59)$$

If  $\lambda$  is desired to more precision than offered by the use of  $\lambda_1$ , then it is merely a matter of applying Eq. (4.58). However, it is evident from Eq. (4.58) that the position of the satellite with respect to the Earth at time  $t_k - \lambda$  is required.



Typically, the time,  $t_k$ , at which the photons are received by the telescope is known, but not the time at which the photons were reflected from the spacecraft, which precedes  $t_k$ . For this reason, when using the convergence method of Eq. (4.58) it is necessary to propagate the satellite position backwards in time by an amount of  $\lambda_i$ . Since the time scales involved with the calculation of  $\lambda$  are quite small, the dominant effect on the change in position over that time period is the central body gravitational acceleration, and more importantly the point mass component of the gravitational acceleration. Therefore, to propagate the satellite position backwards by  $\lambda_i$ , we simply employ numerical integration of a simplified set of equations of motion utilizing only the point mass component of the central body gravitational acceleration. On the other hand, if only an approximation to  $\lambda$  is required and Eq. (4.59) is used, then no numerical integration is required in the computation of  $\lambda$ , but since the position of the satellite at time  $t_k - \lambda$  is needed, this can be obtained by numerically integrating the simplified equations of motion backwards in time by  $\lambda$ .

Once the light time correction has been applied to obtain the satellite location at the time the photons were reflected, the line-of-sight vector at time  $t_k$  from the telescope to the satellite is given by

$$\mathbf{u}^i(t_k) = \frac{\mathbf{r}^i(t_k - \lambda) - \mathbf{r}_{\text{stn}}^i(t_k)}{\|\mathbf{r}^i(t_k - \lambda) - \mathbf{r}_{\text{stn}}^i(t_k)\|}. \quad (4.60)$$

#### 4.2.2 Stellar Aberration Correction

Stellar aberration is the apparent shift in the direction of incoming light due to the velocity of the observer. In our case, this means that since the telescope is

moving (whether it be on the Earth or in orbit) there is a change in the direction of the light coming from a satellite due to the motion of the telescope. This is often referred to as the “raindrop effect” in which raindrops appear to be coming from a different direction than they actually are while you are in motion. The apparent line-of-sight of the satellite at time  $t_k$  is given by<sup>45,47</sup>

$$\mathbf{u}_{\text{app}}^i(t_k) = \frac{\mathbf{u}^i(t_k) + \frac{1}{c}\mathbf{v}_{\text{stn}}^i(t_k)}{\left\| \mathbf{u}^i(t_k) + \frac{1}{c}\mathbf{v}_{\text{stn}}^i(t_k) \right\|}, \quad (4.61)$$

where  $\mathbf{u}^i(t_k)$  is the light time corrected line-of-sight at the time photons were received at the telescope which is given by Eq. (4.60),  $c$  is the speed of light, and  $\mathbf{v}_{\text{stn}}^i(t_k)$  is the velocity of the telescope at the time the photons were received at the telescope. The velocity of the telescope (assuming it to be ground-based) is

$$\mathbf{v}_{\text{stn}}^i(t_k) = \mathbf{T}_f^i(t_k)(\boldsymbol{\omega}_{f/i}^f \times \mathbf{r}_{\text{stn}}^f),$$

where  $\mathbf{T}_f^i(t_k)$  is the orientation of the Earth-fixed reference frame with respect to the inertial reference frame at time  $t_k$ ,  $\boldsymbol{\omega}_{f/i}^f$  is the angular velocity of the Earth with respect to the inertial reference frame, and  $\mathbf{r}_{\text{stn}}^f$  is the location of the telescope in the Earth-fixed reference frame.

### 4.2.3 Lighting Conditions

In order for line-of-sight measurements to be taken, two lighting conditions must be met: the satellite must be in sunlight at the time when photons are reflected off of the satellite (that is at time  $t_k - \lambda$ ) and the observer must be in shadow at the time when the reflected photons are received at the telescope (that is at time  $t_k$ ).

#### 4.2.3.1 Satellite Lighting Condition

As the Sun moves behind the Earth (i.e. as the satellite moves into the penumbra), the geometry is defined as shown in Figure 4.2. Let the radius of the Earth be  $R_e$ , the radius of the Sun be  $R_s$ , the position of the Sun with respect to the Earth at the time at which photons left the Sun be  $\mathbf{r}_{\text{sun}}^i(t_k - \lambda_s)$  where  $\lambda_s$  is the one-way light time from the Sun to the Earth, and the position of the satellite with respect to the Earth at the time at which photons were reflected off of the satellite be  $\mathbf{r}^i(t_k - \lambda)$ . It can be shown that  $\gamma$ ,  $\tau$ , and  $\epsilon$  as defined in Figure 4.2 are given by<sup>19</sup>

$$\gamma = \sin^{-1} \frac{R_e}{\|\mathbf{r}^i(t_k - \lambda)\|}, \quad \tau = \sin^{-1} \frac{R_s}{\|\mathbf{r}_{\text{sun}}^i(t_k - \lambda_s) - \mathbf{r}^i(t_k - \lambda)\|} \quad \text{and}$$

$$\epsilon = \cos^{-1} \frac{[\mathbf{r}_{\text{sun}}^i(t_k - \lambda_s) - \mathbf{r}^i(t_k - \lambda)] \cdot \mathbf{r}^i(t_k - \lambda)}{\|\mathbf{r}_{\text{sun}}^i(t_k - \lambda_s) - \mathbf{r}^i(t_k - \lambda)\| \|\mathbf{r}^i(t_k - \lambda)\|}.$$

Therefore, by Figure 4.2, if

$$\epsilon > \gamma + \tau,$$

then the satellite was in full sunlight at time  $t_k - \lambda$ . It may be that photons are also reflected for some period during the time when the satellite is in the penumbra of the Earth; however, it is sufficient to assume that measurements are only generated during full sunlight for the current work.

#### 4.2.3.2 Observer Lighting Condition

Similar to the case of determining if the satellite was in full sunlight, we must also determine if the telescope was in shadow at the time of reception of the photons (that is at time  $t_k$ ). For this case, we apply the same analysis as for the satellite

except that we are now interested in the telescope being in umbra. Let the radius of the Earth be  $R_e$ , the radius of the Sun be  $R_s$ , the position of the Sun with respect to the Earth at the time at which photons left the Sun be  $\mathbf{r}_{\text{sun}}^i(t_k - \lambda_s)$ , and the position of the telescope with respect to the Earth be at the time at which photons were received at the telescope be  $\mathbf{r}_{\text{stn}}^i(t_k)$ . It can be shown that  $\gamma$ ,  $\tau$ , and  $\epsilon$  as defined in Figure 4.2 are given by<sup>19</sup>

$$\begin{aligned} \gamma &= \sin^{-1} \frac{R_e}{\|\mathbf{r}_{\text{stn}}^i(t_k)\|}, & \tau &= \sin^{-1} \frac{R_s}{\|\mathbf{r}_{\text{sun}}^i(t_k - \lambda_s) - \mathbf{r}_{\text{stn}}^i(t_k)\|} & \text{and} \\ \epsilon &= \cos^{-1} \frac{[\mathbf{r}_{\text{sun}}^i(t_k - \lambda_s) - \mathbf{r}_{\text{stn}}^i(t_k)] \cdot \mathbf{r}_{\text{stn}}^i(t_k)}{\|\mathbf{r}_{\text{sun}}^i(t_k - \lambda_s) - \mathbf{r}_{\text{stn}}^i(t_k)\| \|\mathbf{r}_{\text{stn}}^i(t_k)\|}. \end{aligned}$$

It suffices to determine if the station was in umbra at the time photons were received at the telescope, that is at time  $t_k$ . This case is met provided that

$$\tau + \epsilon < \gamma,$$

as is readily observed from Figure 4.2.

#### 4.2.4 Field of View Condition

One final condition must be checked to determine if line-of-sight measurements are to be taken at time  $t_k$ . That condition is that the light time corrected line-of-sight given in Eq. (4.60) must have been in the field of view of the telescope at time  $t_k$ . In order to compute the field of view condition, it is necessary to determine the focal plane angles of the line-of-sight vector and determine if these are within the field of view of the telescope. This, in turn, requires us to know the pointing of the telescope in order to compute the focal plane angles. As such, we assume that the

orientation of the sensor platform (denoted hereafter by  $p$ ) is known with respect to a local surface frame (denoted hereafter by  $s$ ) at time  $t_k$ . Then, the line-of-sight vector in the telescope reference frame is given by a coordinate transformation of Eq. (4.60), that is

$$\mathbf{u}^s(t_k) = \mathbf{T}_s^p(t_k) \mathbf{T}_f^s \mathbf{T}_i^f(t_k) \mathbf{u}^i(t_k). \quad (4.62)$$

It is worth noting that in Eq. (4.62) while the surface-to-platform and inertial-to-fixed transformations are time dependent, the fixed-to-surface transformation is not; this is due to the fact that the two reference frames are fixed (in time) with respect to one another and the transformation depends only on the latitude and longitude of the telescope. From Eq. (4.62) we can extract focal plane angles as

$$\zeta = \tan^{-1} \frac{u_x}{u_z} \quad \text{and} \quad \eta = \tan^{-1} \frac{u_y}{u_z},$$

where  $\mathbf{u}^s(t_k) = [u_x \ u_y \ u_z]^T$ . Now, assume that the field of view of the telescope is defined by  $\zeta_{\max}$  and  $\eta_{\max}$ , which represent the maximum values of  $\zeta$  and  $\eta$  which allow the line-of-sight to be within view of the telescope. Then, if  $|\zeta| \leq \zeta_{\max}$  and  $|\eta| \leq \eta_{\max}$ , the satellite is within the field of view of the telescope, and line-of-sight measurements are allowed.

#### 4.2.5 Right Ascension and Declination Measurements

Provided that the lighting conditions of Section 4.2.3 and the field of view condition of Section 4.2.4 are met, then the right ascension and declination are readily computed from the apparent line-of-sight in Eq. (4.61) as

$$\alpha_k = \tan^{-1} \frac{u_{\text{app},y}}{u_{\text{app},x}} \quad \text{and} \quad \delta_k = \tan^{-1} \frac{u_{\text{app},z}}{d_{\text{app}}}, \quad (4.63)$$

where  $\mathbf{u}_{\text{app}}^i(t_k) = [u_{\text{app},x} \ u_{\text{app},y} \ u_{\text{app},z}]^T$  and  $d_{\text{app}}^2 = u_{\text{app},x}^2 + u_{\text{app},y}^2$ . Then, concatenating the right ascension and declination at time  $t_k$ , given respectively by  $\alpha_k$  and  $\delta_k$ , we can formulate the measurement of right ascension and declination at time  $t_k$  as

$$\mathbf{y}_k = \begin{bmatrix} \alpha_k \\ \delta_k \end{bmatrix} + \mathbf{v}_{\alpha\delta,k}, \quad (4.64)$$

where  $\mathbf{v}_{\alpha\delta,k}$  is the measurement noise which is assumed to be a white-noise sequence with mean and covariance

$$\mathbb{E}\{\mathbf{v}_{\alpha\delta,k}\} = \mathbf{0} \quad \forall \quad k \quad \text{and} \quad \mathbb{E}\{\mathbf{v}_{\alpha\delta,k} \mathbf{v}_{\alpha\delta,k'}^T\} = \mathbf{R}_{\alpha\delta,k} \delta_{k,k'},$$

with  $\delta_{k,k'}$  representing the Kronecker delta.

In computing the derivatives of the right ascension and declination, we do not need to consider the lighting conditions or the field of view condition since these serve only to determine if measurements were taken at time  $t_k$ . That is, when processing measurements of right ascension and declination, these conditions are not checked since they would have been met in generating the measurements.

To compute the derivative of the right ascension and declination with respect to the satellite position, we utilize the chain rule applied to Eq. (4.64) and find that

$$\frac{\partial \mathbf{y}_k}{\partial \mathbf{r}^i(t_k)} = \frac{\partial}{\partial \mathbf{u}_{\text{app}}^i} \left\{ \begin{bmatrix} \alpha_k \\ \delta_k \end{bmatrix} \right\} \frac{\partial \mathbf{u}_{\text{app}}^i(t_k)}{\partial \mathbf{u}^i(t_k)} \frac{\partial \mathbf{u}^i(t_k)}{\partial \mathbf{r}^i(t_k - \lambda)} \frac{\partial \mathbf{r}^i(t_k - \lambda)}{\partial \mathbf{r}^i(t_k)}. \quad (4.65)$$

From Eq. (4.63), it can be shown that the first derivative term in Eq. (4.65) is given by

$$\mathbf{U} = \frac{\partial}{\partial \mathbf{u}_{\text{app}}^i} \left\{ \begin{bmatrix} \alpha_k \\ \delta_k \end{bmatrix} \right\} = \begin{bmatrix} \frac{-u_{\text{app},y}}{d_{\text{app}}^2} & \frac{u_{\text{app},x}}{d_{\text{app}}^2} & 0 \\ \frac{-u_{\text{app},x}u_{\text{app},z}}{d_{\text{app}}} & \frac{-u_{\text{app},y}u_{\text{app},z}}{d_{\text{app}}} & d_{\text{app}} \end{bmatrix}. \quad (4.66)$$

Similarly, the second and third derivative terms in Eq. (4.65) are readily found to be

$$\frac{\partial \mathbf{u}_{\text{app}}^i(t_k)}{\partial \mathbf{u}^i(t_k)} = -\frac{1}{\|\mathbf{u}^i(t_k) + \frac{1}{c}\mathbf{v}_{\text{stn}}^i(t_k)\|}[\mathbf{u}_{\text{app}}^i(t_k) \times]^2 \quad (4.67a)$$

$$\frac{\partial \mathbf{u}^i(t_k)}{\partial \mathbf{r}^i(t_k - \lambda)} = -\frac{1}{\|\mathbf{r}^i(t_k - \lambda) - \mathbf{r}_{\text{stn}}^i(t_k)\|}[\mathbf{u}^i(t_k) \times]^2. \quad (4.67b)$$

The final derivative term in Eq. (4.65) is related to the state transition matrix, which maps variations across time, that is, the state transition matrix is defined as

$$\Phi(t_k, t_m) = \frac{\partial \mathbf{x}(t_k)}{\partial \mathbf{x}(t_m)}.$$

The final derivative term in Eq. (4.65) is the upper  $3 \times 3$  block of  $\Phi(t_k - \lambda, t_k)$  provided that the states are ordered with the satellite position as the first 3 elements. Furthermore, the state transition matrix satisfies the differential equation

$$\dot{\Phi}(\sigma, t_k) = \mathbf{F}(\sigma)\Phi(\sigma, t_k), \quad (4.68)$$

where  $\mathbf{F}(\sigma)$  represents the matrix of first derivatives of the nonlinear dynamics (i.e. the Jacobian matrix) which in this case is used to relate errors in the state at time  $t_m$  to errors in the state at time  $t_k$ . Furthermore, we let  $\sigma$  be related to the running time variable,  $t$  via  $\sigma = t - \lambda$  and we consider the range of  $t$  as  $t \in [t_k + \lambda, t_k]$ . Applying an Euler integration scheme to the state transition matrix differential equation of Eq. (4.68), which is valid since the change in time required is small, it is found that

$$\Phi(t_k - \lambda, t_k) = \Phi(t_k, t_k) + \mathbf{F}(t_k)(t_k - t_k - \lambda) = \mathbf{I} - \lambda \mathbf{F}(t_k). \quad (4.69)$$

Therefore, by the properties of the tangent linear dynamics (i.e.  $\mathbf{F}(t_k)$ ), it is seen that

$$\frac{\partial \mathbf{r}^i(t_k - \lambda)}{\partial \mathbf{r}^i(t_k)} = \mathbf{I}. \quad (4.70)$$

Substituting Eqs. (4.66)–(4.67) and Eq. (4.70) into Eq. (4.65), we find that the derivative of the right ascension and declination measurement with respect to the satellite position is given by

$$\frac{\partial \mathbf{y}_k}{\partial \mathbf{r}^i(t_k)} = \frac{1}{\|\mathbf{u}^i(t_k) + \frac{1}{c}\mathbf{v}_{\text{stn}}^i(t_k)\|} \frac{1}{\|\mathbf{r}^i(t_k - \lambda) - \mathbf{r}_{\text{stn}}^i(t_k)\|} \mathbf{U}[\mathbf{u}_{\text{app}}^i(t_k) \times]^2 [\mathbf{u}^i(t_k) \times]^2.$$

Similar to the derivative with respect to satellite position, to compute the derivative of the right ascension and declination with respect to the satellite velocity, we utilize the chain rule applied to Eq. (4.64) and find that

$$\frac{\partial \mathbf{y}_k}{\partial \mathbf{v}^i(t_k)} = \frac{\partial}{\partial \mathbf{u}_{\text{app}}^i} \left\{ \begin{bmatrix} \alpha_k \\ \delta_k \end{bmatrix} \right\} \frac{\partial \mathbf{u}_{\text{app}}^i(t_k)}{\partial \mathbf{u}^i(t_k)} \frac{\partial \mathbf{u}^i(t_k)}{\partial \mathbf{r}^i(t_k - \lambda)} \frac{\partial \mathbf{r}^i(t_k - \lambda)}{\partial \mathbf{v}^i(t_k)}. \quad (4.71)$$

The first three derivative terms have already been determined and are given by Eqs. (4.66)–(4.67). The final derivative term comes from a similar path as previously taken, and is again related to the state transition matrix  $\Phi(t_k - \lambda, t_k)$ , except that it is the first 3 rows and second set of 3 columns for the satellite velocity, provided that the satellite velocity is the second set of 3 elements in the state vector. Again, we arrive at Eq. (4.69) as the form of the state transition matrix, and by the properties of the tangent linear dynamics, we find that

$$\frac{\partial \mathbf{r}^i(t_k - \lambda)}{\partial \mathbf{v}^i(t_k)} = -\lambda \mathbf{I}. \quad (4.72)$$

Therefore, substituting Eqs. (4.66)–(4.67) and Eq. (4.72) into Eq. (4.71), we find that the derivative of the right ascension and declination measurement with respect to the satellite velocity is given by

$$\frac{\partial \mathbf{y}_k}{\partial \mathbf{v}^i(t_k)} = -\lambda \frac{1}{\|\mathbf{u}^i(t_k) + \frac{1}{c}\mathbf{v}_{\text{stn}}^i(t_k)\|} \frac{1}{\|\mathbf{r}^i(t_k - \lambda) - \mathbf{r}_{\text{stn}}^i(t_k)\|} \mathbf{U}[\mathbf{u}_{\text{app}}^i(t_k) \times]^2 [\mathbf{u}^i(t_k) \times]^2.$$



## Chapter 5

### Results

To evaluate the performance of the proposed splitting Gaussian mixture unscented Kalman filter (SGMUKF) algorithm with respect to the more standard extended Kalman filter (EKF) and unscented Kalman filter (UKF) algorithms, the problem of tracking a resident space object (RSO) in a near-geosynchronous orbit is considered. The models for the nonlinear dynamical system which describes the time-evolution of the position, velocity, attitude, and angular velocity of an RSO were presented in Chapter 4. Similarly, the models representing the observational relationships which describe the measurement of the line-of-sight of an RSO from a ground station were also presented in Chapter 4. In order to systematically approach the evaluation of the SGMUKF, a simplified tracking model is first considered. As was shown in Cook,<sup>9</sup> the most dominant spacecraft acceleration in a geosynchronous orbit is that of the point mass gravitational acceleration. This is demonstrated via a summary of the typical spacecraft accelerations given in Table 5.1, which shows that for an object with an area-to-mass ratio of  $0.01\text{m}^2/\text{kg}$  that the point mass gravitational acceleration is several orders of magnitude larger than any other common acceleration.

Table 5.1: Magnitude of Typical Spacecraft Accelerations<sup>9</sup>

Perturbation	Acceleration (in m/s <sup>2</sup> ) for Geosynchronous Spacecraft with $A/m = 0.01\text{m}^2/\text{kg}$
Earth Point Mass	$2.2 \times 10^{-1}$
Earth's Oblateness ( $J_2$ )	$7.4 \times 10^{-6}$
Lunar Third Body	$7.3 \times 10^{-6}$
Solar Third Body	$3.3 \times 10^{-6}$
Solar Radiation Pressure	$4.6 \times 10^{-8}$

Therefore, before considering the full tracking model as described by the models in Chapter 4, we first consider a simplified tracking model as applied to the propagation of uncertainty for a circular and an eccentric planar orbit. We then proceed to consideration of the propagation of uncertainty in the full tracking model, and finally conclude with the implementation of the inclusion of measurement data for the update in the full tracking model.

## 5.1 Propagation in a Simplified Tracking Model

For the simplified tracking model, the rotational motion of the vehicle is neglected and the only active acceleration modeled is that of the central body gravity. Making these adjustments to the full tracking model equations of motion yields the simplified tracking model equations of motion to be

$$\dot{\mathbf{r}}^i = \mathbf{v}^i$$

$$\dot{\mathbf{v}}^i = \mathbf{a}_g^i(\mathbf{r}^i),$$

where  $\mathbf{a}_g^i(\mathbf{r}^i)$  is the central body acceleration, which is modeled by the point mass approximation, given by

$$\mathbf{a}_g^i = -\frac{\mu}{r^3} \mathbf{r}^i.$$

Furthermore, the simplified tracking model assumes that the motion of the vehicle is confined to the equatorial plane, which allows the position to be described by two scalar values,  $x$  and  $y$ , and the velocity to be described by two scalar values  $\dot{x}$  and  $\dot{y}$ . Therefore, the state vector and equations of motion that describe the nonlinear dynamical system are

$$\mathbf{x}(t) = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} \quad \text{and} \quad \mathbf{f}(\mathbf{x}(t), t) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ -\frac{\mu x}{r^3} \\ -\frac{\mu y}{r^3} \end{bmatrix},$$

where  $\mu$  is the gravitational constant of the central body, and  $r = \sqrt{x^2 + y^2}$  is the distance from the central body to the vehicle. The linearized dynamics Jacobian is also modified from the full tracking model to yield the simplified tracking model Jacobian as

$$\mathbf{F}(\mathbf{x}(t), t) = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{A}_g & \mathbf{0} \end{bmatrix}, \quad (5.1)$$

where

$$\mathbf{A}_g = \frac{\partial \mathbf{a}_g^i}{\partial \mathbf{r}^i} = \frac{\mu}{r^5} (3\mathbf{r}^i(\mathbf{r}^i)^T - (\mathbf{r}^i)^T \mathbf{r}^i \mathbf{I})$$

By again confining the motion to the equatorial plane,  $\mathbf{A}_g$  may be written explicitly in terms of the state variables as

$$\mathbf{A}_g = \frac{\mu}{r^3} \begin{bmatrix} 3 \left(\frac{x}{r}\right)^2 - 1 & 3 \frac{x y}{r^2} \\ 3 \frac{x y}{r^2} & 3 \left(\frac{y}{r}\right)^2 - 1 \end{bmatrix}.$$

Finally, since the SGMUKF recursive filtering scheme relies on the implementation of either differential entropy or Rényi entropy, it is worth noting that the trace of the linearized dynamics Jacobian for the simplified tracking model is zero, which is readily verified by inspection of Eq. (5.1). This means that the differential entropy and the Rényi entropy are constant for the linearized dynamical system, which simplifies the implementation of the SGMUKF by allowing the predicted entropy to be compared against some reference value without needing to implement a differential equation to solve for the entropy of the linearized system.

Two orbits are considered for testing the SGMUKF method, with the first orbit characterized by a semi-major axis of 42000 [km] and zero eccentricity and the second orbit characterized by a semi-major axis of 35000 [km] and an eccentricity of 0.2. The second orbit's eccentricity was chosen so that the orbit has an apoapse of 42000 [km]. The nominal orbits are shown in Figure 5.1.

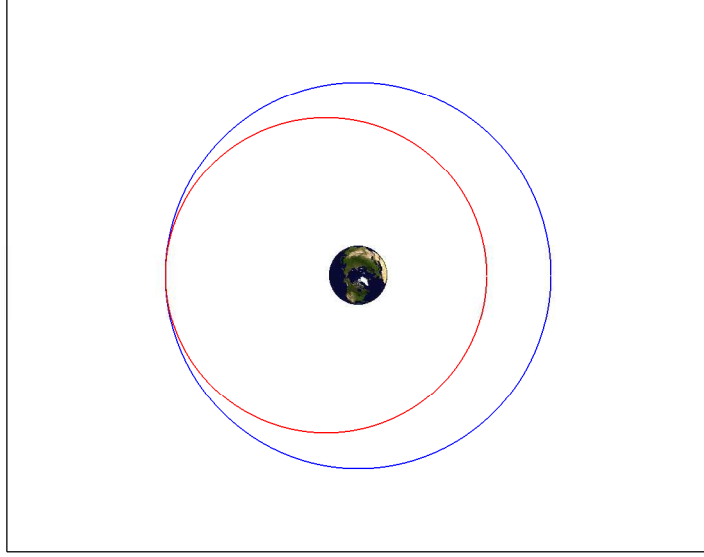


Figure 5.1: Nominal trajectories for the circular (in blue) and eccentric (in red) test cases in the simplified tracking model.

In each case, the initial uncertainty on the position is taken to be 1 [km], and the initial uncertainty on the velocity is taken to be 1 [m/s], such that the initial covariance is given by

$$\mathbf{P}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 \times 10^{-6} & 0 \\ 0 & 0 & 0 & 1 \times 10^{-6} \end{bmatrix},$$

with the units being  $\text{km}^2$  and  $(\text{km/s})^2$  for the position and velocity coordinates, respectively.

### 5.1.1 Circular Orbit Test Case

To provide a relative measure of the performance of the filters, the likelihood agreement measure is computed using samples from a monte carlo simulation and the predicted probability density functions (pdfs) from the EKF, UKF, and SGMUKF. The likelihood measures of the EKF and UKF are normalized by the value for the SGMUKF so as to provide a relative measure with respect to the SGMUKF; that is, if the normalized likelihood of the EKF or UKF were to exceed unity, it would be better performing than the SGMUKF. Unfortunately, the covariance for the EKF becomes ill-conditioned with respect to matrix inversion within a short period of time; therefore, its likelihood agreement measure cannot be computed and it is excluded in the results. However, should the EKF not be near-singular, the analysis would be similar to that of the UKF. This analysis is summarized in Figure 5.2, wherein it can be observed that the UKF is clearly outperformed by the SGMUKF. The rapid departure of the likelihood agreement of the UKF from that of the SGMUKF which occurs after approximate 12 hours of propagation is the same point at which the SGMUKF first detects nonlinearity in the propagation and begins the process of splitting.

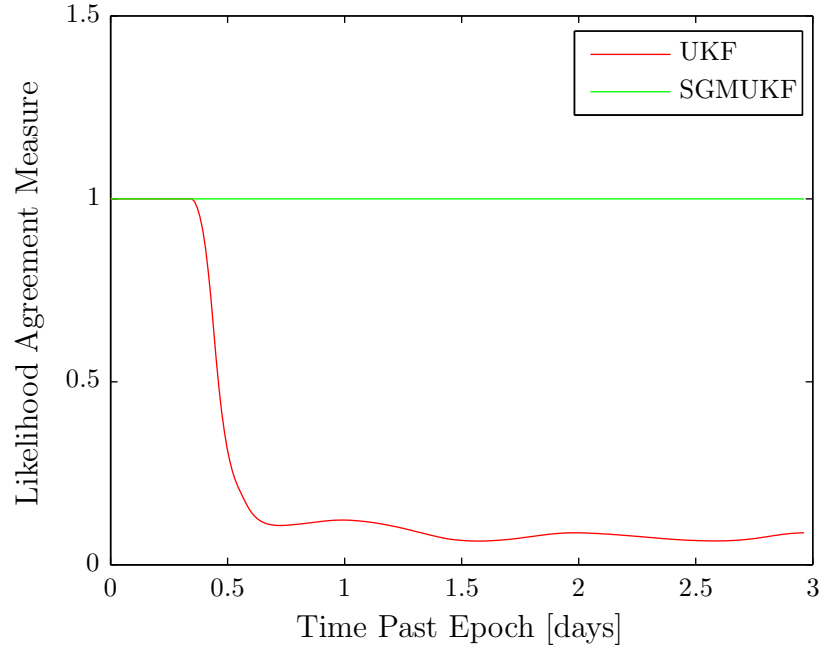


Figure 5.2: Likelihood agreement measure for the UKF and SGMUKF, normalized by the value for the SGMUKF

Figures 5.3–5.10 show the position and velocity pdf contours at four times: the initial time, one time period of the nominal orbit, two time periods of the nominal orbit, and three time periods of the nominal orbit. In each figure, the pdf contours are shown for the EKF, the UKF, and the SGMUKF methods along with samples derived from a monte carlo simulation, which is run by propagating samples drawn from the initial distribution. Figures 5.3 and 5.4 show the position and velocity pdf contours at the initial time, and can be seen to be identical for each filtering method since all of the filters are initialized with the same mean and covariance. Figures 5.5 and 5.6 show the position and velocity pdf contours after one time period of the nominal circular

orbit. At this point, the SGMUKF has already begun its splitting process, enabling it to better map the curvature exhibited by the monte carlo samples. Furthermore, while both the EKF and UKF cannot achieve the curvature shown by the SGMUKF, it can be seen that the UKF has contours which capture more of the monte carlo samples than the EKF. The better representation of the monte carlo samples by the UKF is further shown in Figures 5.7 and 5.8, where it can be seen that the EKF contours have become even thinner, leading to a poorer representation of the monte carlo samples. The SGMUKF has continued evolving into more components via the splitting process, and continues its matching of the curvature of the samples. Finally, Figures 5.9 and 5.10 show the position and velocity contours at three time periods of the nominal circular orbit, wherein the previous described characteristics have continued to become even more pronounced.



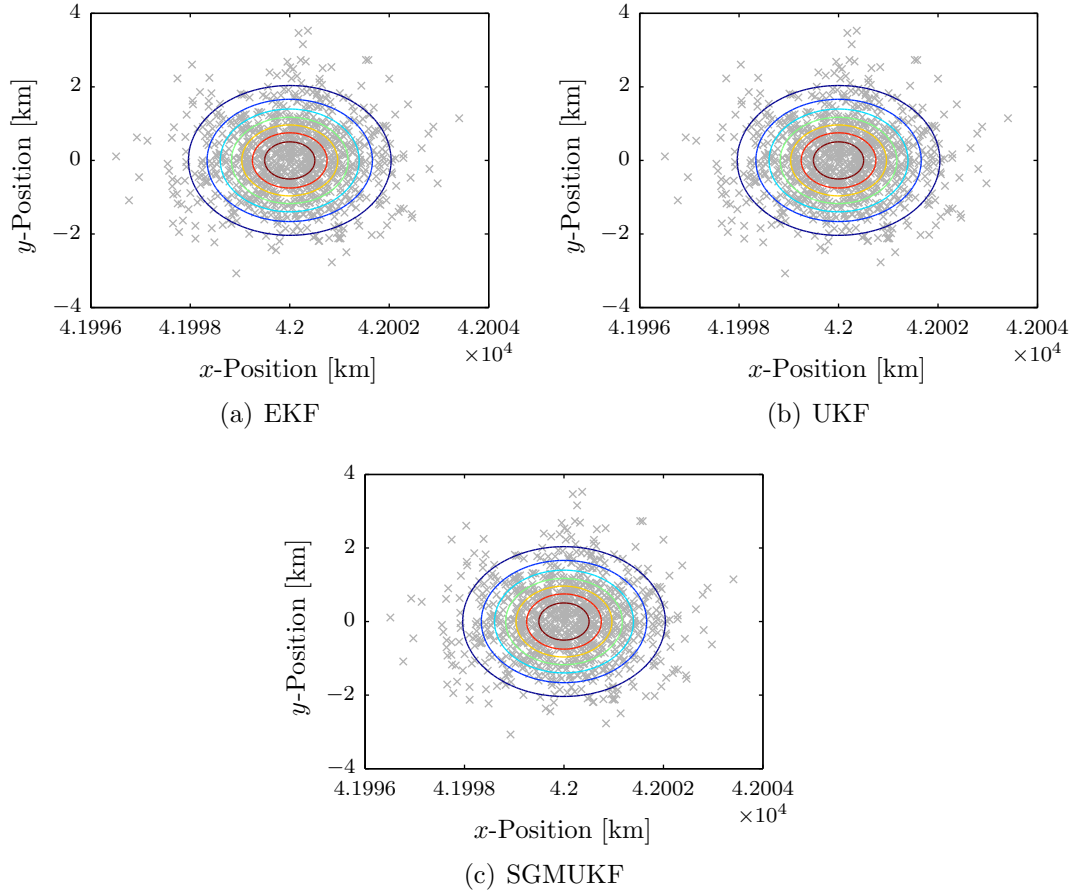


Figure 5.3: Position pdf contours with monte carlo samples at epoch.

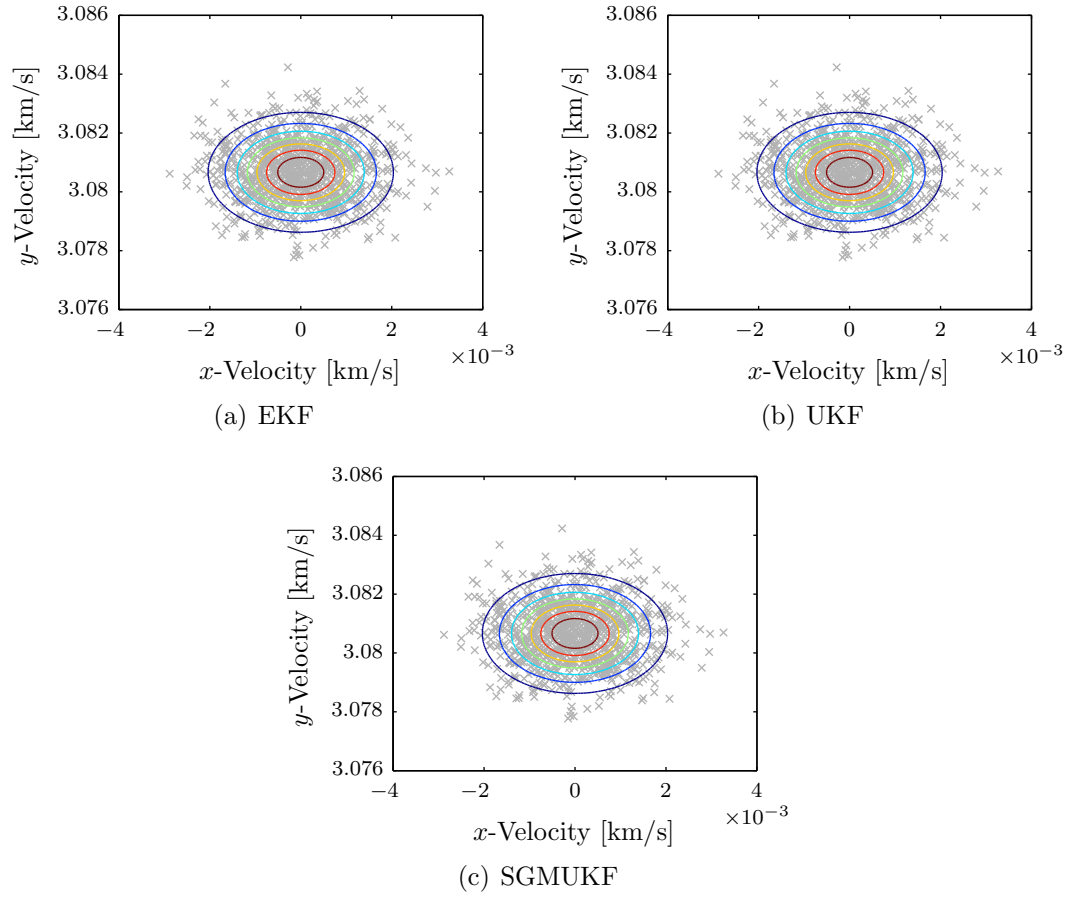


Figure 5.4: Velocity pdf contours with monte carlo samples at epoch.

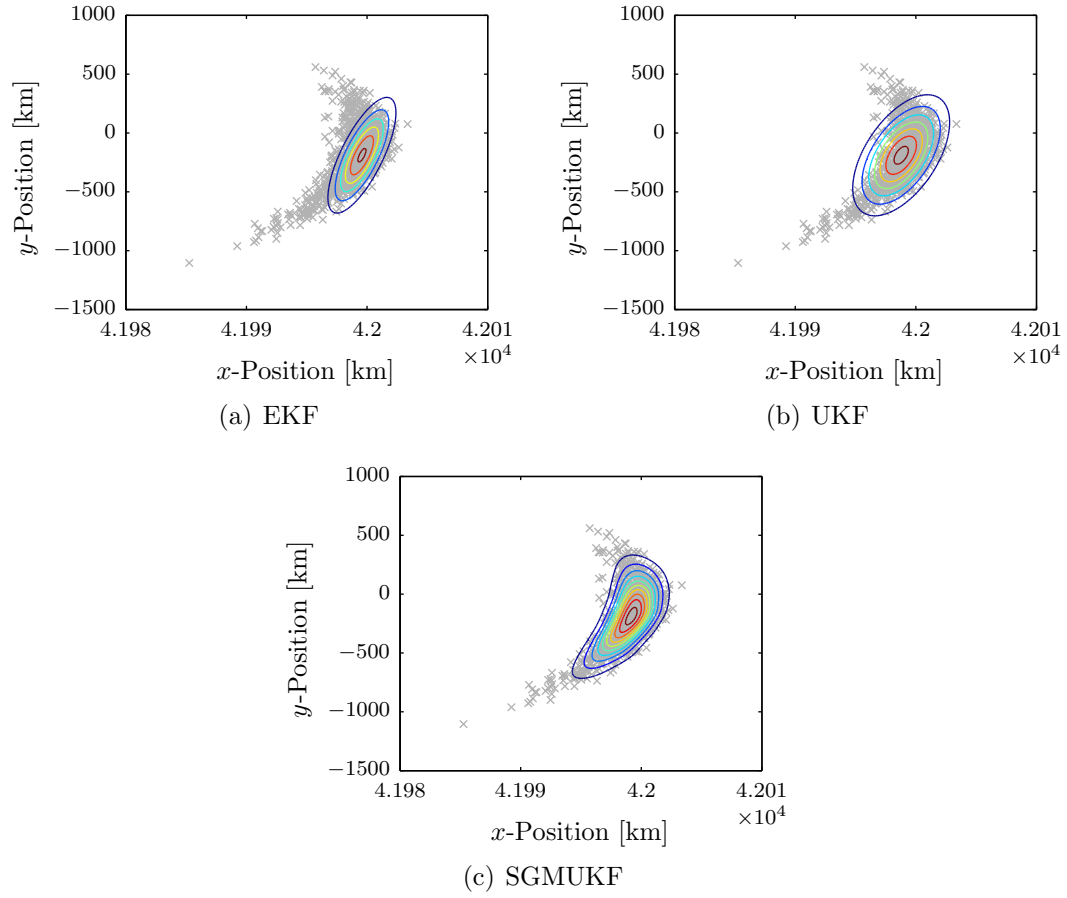


Figure 5.5: Position pdf contours with monte carlo samples at one period of the nominal orbit.

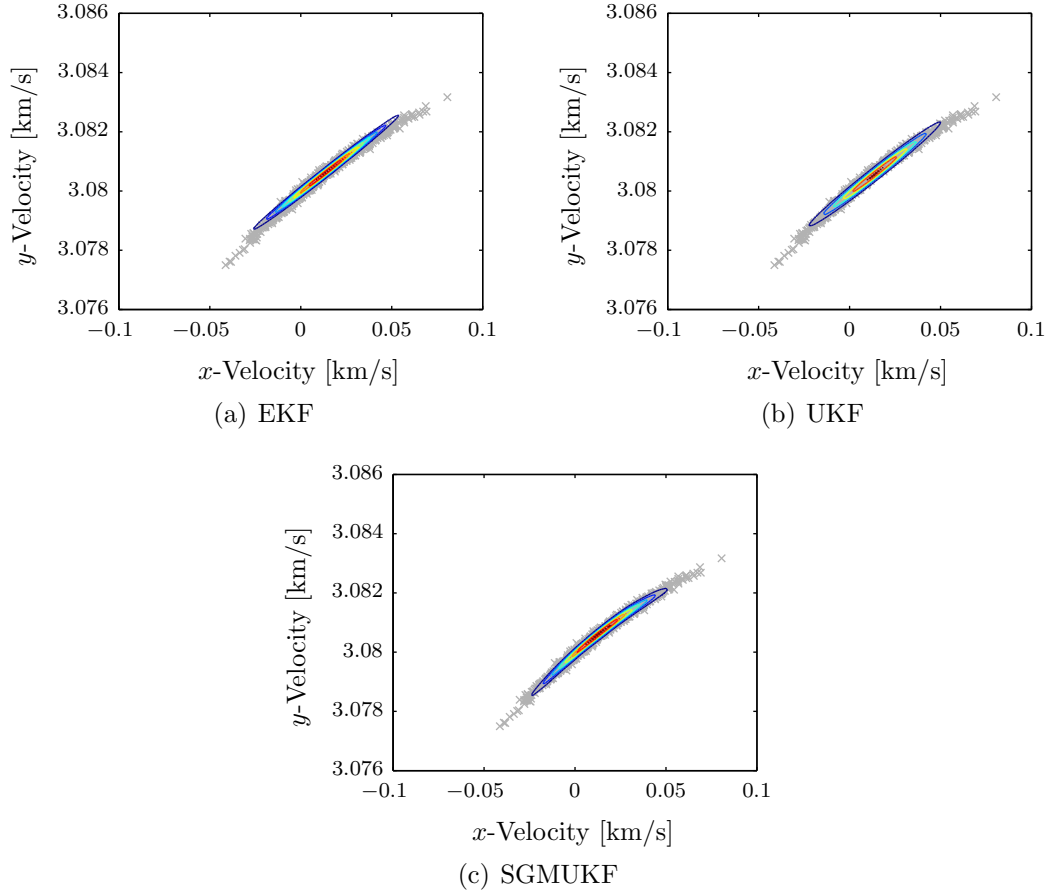


Figure 5.6: Velocity pdf contours with monte carlo samples at one period of the nominal orbit.

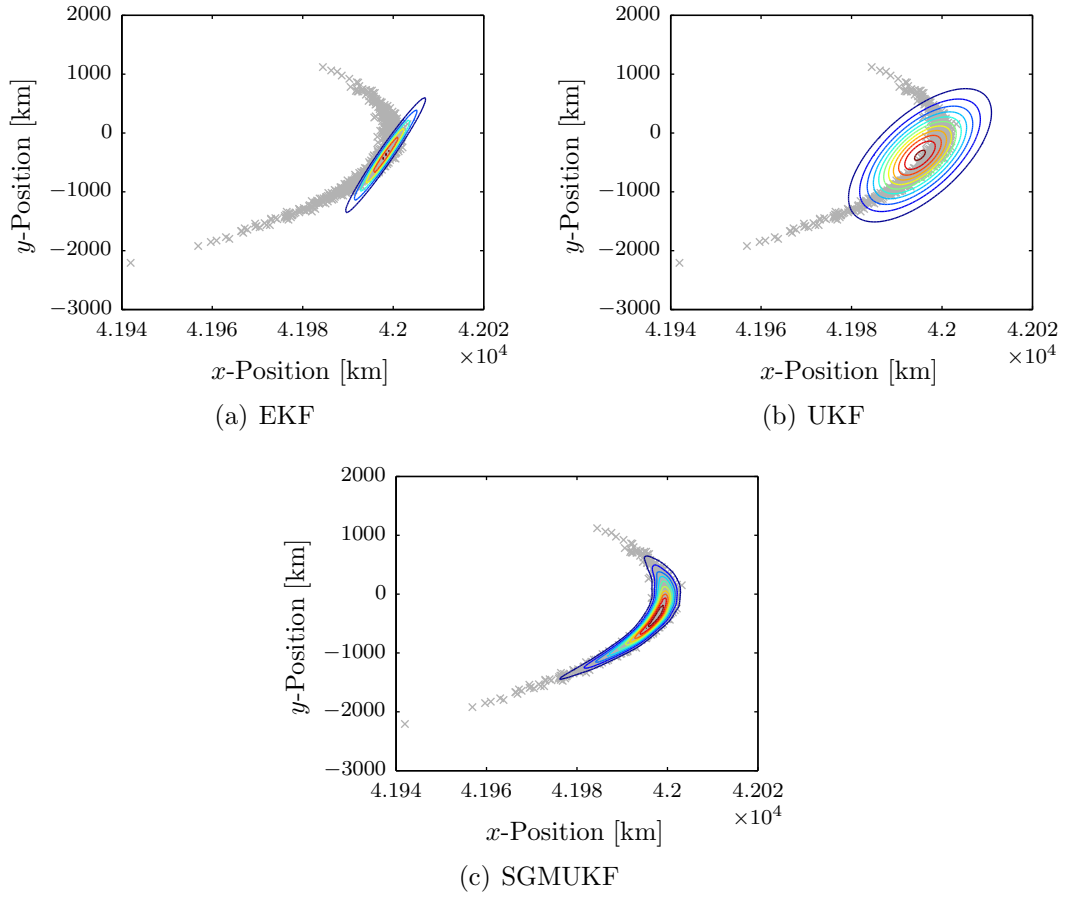


Figure 5.7: Position pdf contours with monte carlo samples at two periods of the nominal orbit.

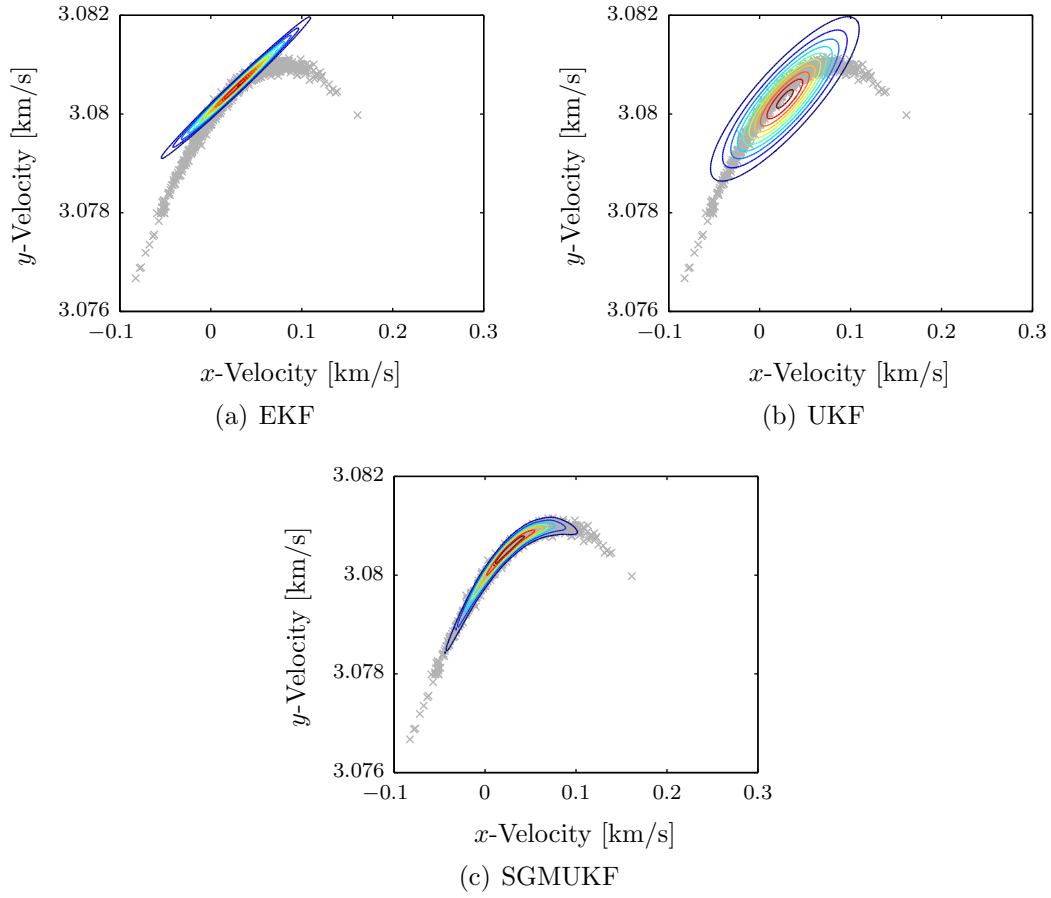


Figure 5.8: Velocity pdf contours with monte carlo samples at two periods of the nominal orbit.

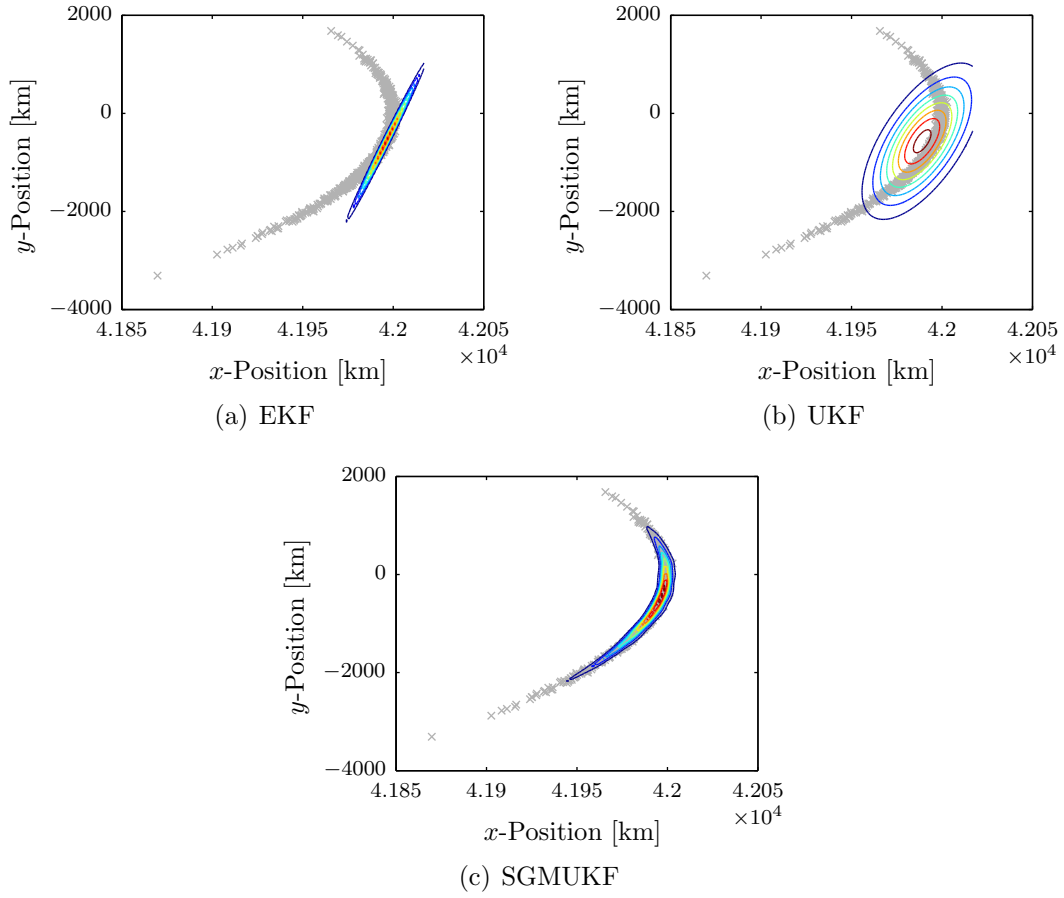


Figure 5.9: Position pdf contours with monte carlo samples at three periods of the nominal orbit.

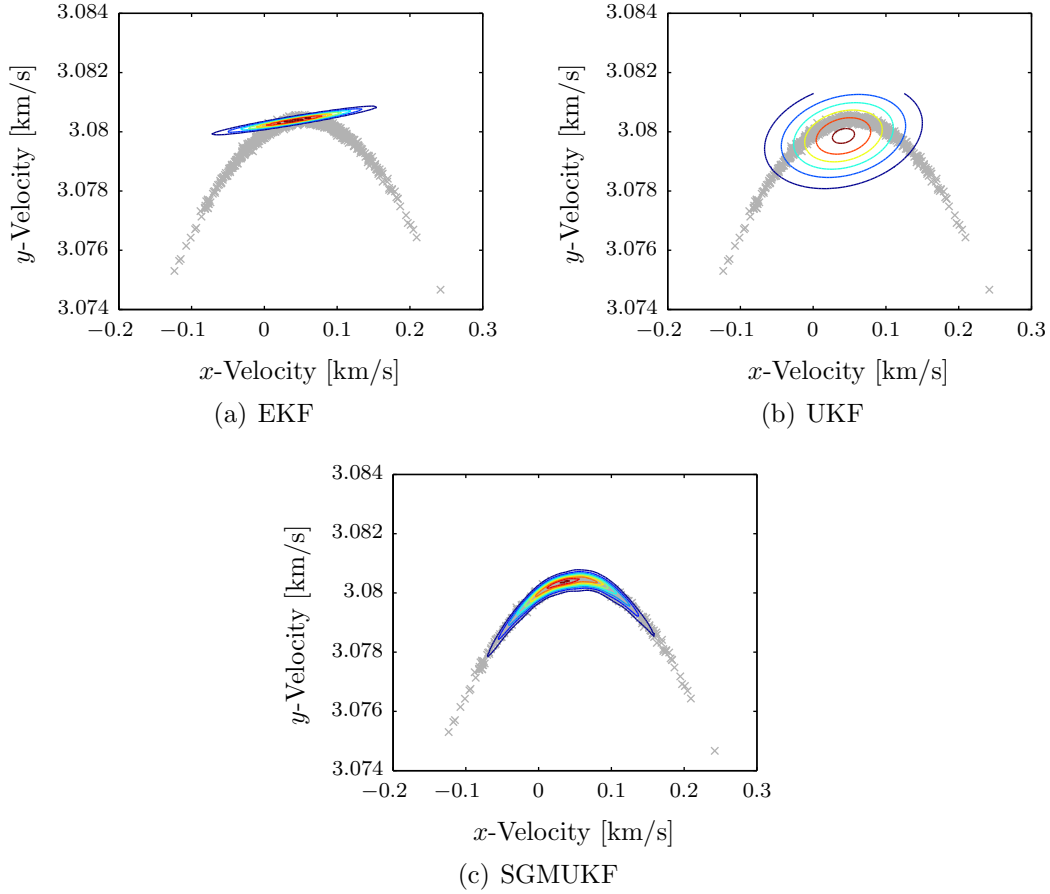


Figure 5.10: Velocity pdf contours with monte carlo samples at three periods of the nominal orbit.

### 5.1.2 Eccentric Orbit Test Case

As was done with the circular orbit test case, a relative measure of the performance of the filters is computed via the likelihood agreement measure, which is determined using the samples from a monte carlo simulation and the predicted pdfs from the EKF, UKF, and SGMUKF. The likelihood measures of the EKF and UKF



are normalized by the value for the SGMUKF so as to provide a relative measure with respect to the SGMUKF. Similar to the previous test case, the EKF covariance becomes ill-conditioned with respect to matrix inversion within a short period of time, once again rendering the likelihood agreement measure for the EKF incalculable. The normalized likelihood agreement measure for the UKF and SGMUKF are shown in Figure 5.11, which shows the UKF being very clearly outperformed again by the SGMUKF.

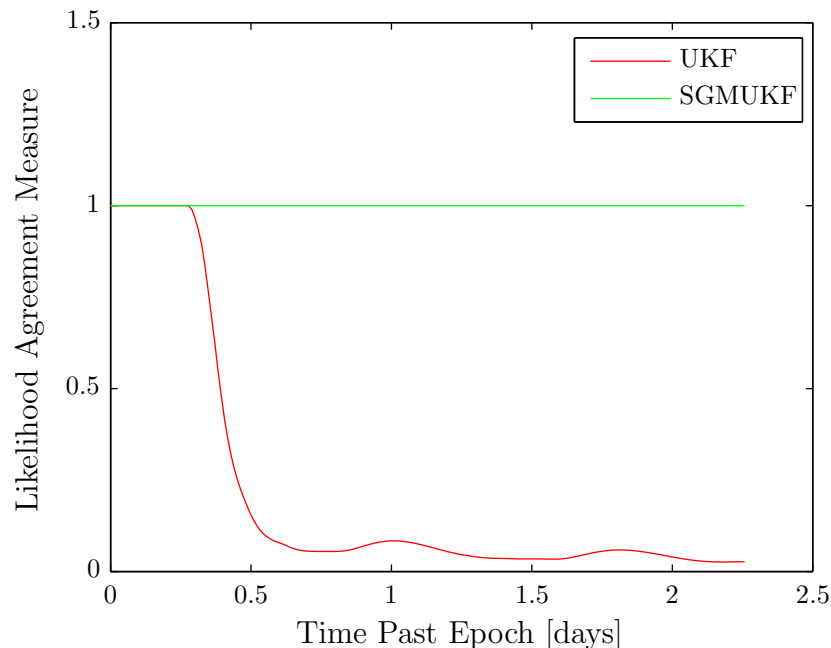


Figure 5.11: Likelihood agreement measure for the UKF and SGMUKF, normalized by the value for the SGMUKF

Figures 5.12–5.19 show the position and velocity pdf contours at four times: the initial time, one time period of the nominal orbit, two time periods of the nominal

orbit, and three time periods of the nominal orbit. In each figure, the pdf contours are shown for the EKF, the UKF, and the SGMUKF methods along with samples derived from a monte carlo simulation, which is run by propagating samples drawn from the initial distribution. In the same manner as described in the circular orbit test case, it can be seen that as the time progresses, the EKF contours become increasingly less representative of the monte carlo samples, the UKF contours manage to capture some number of the monte carlo samples but are not able to match the curvature of the samples, and the SGMUKF contours are able to match the curvature of the samples through the splitting process.

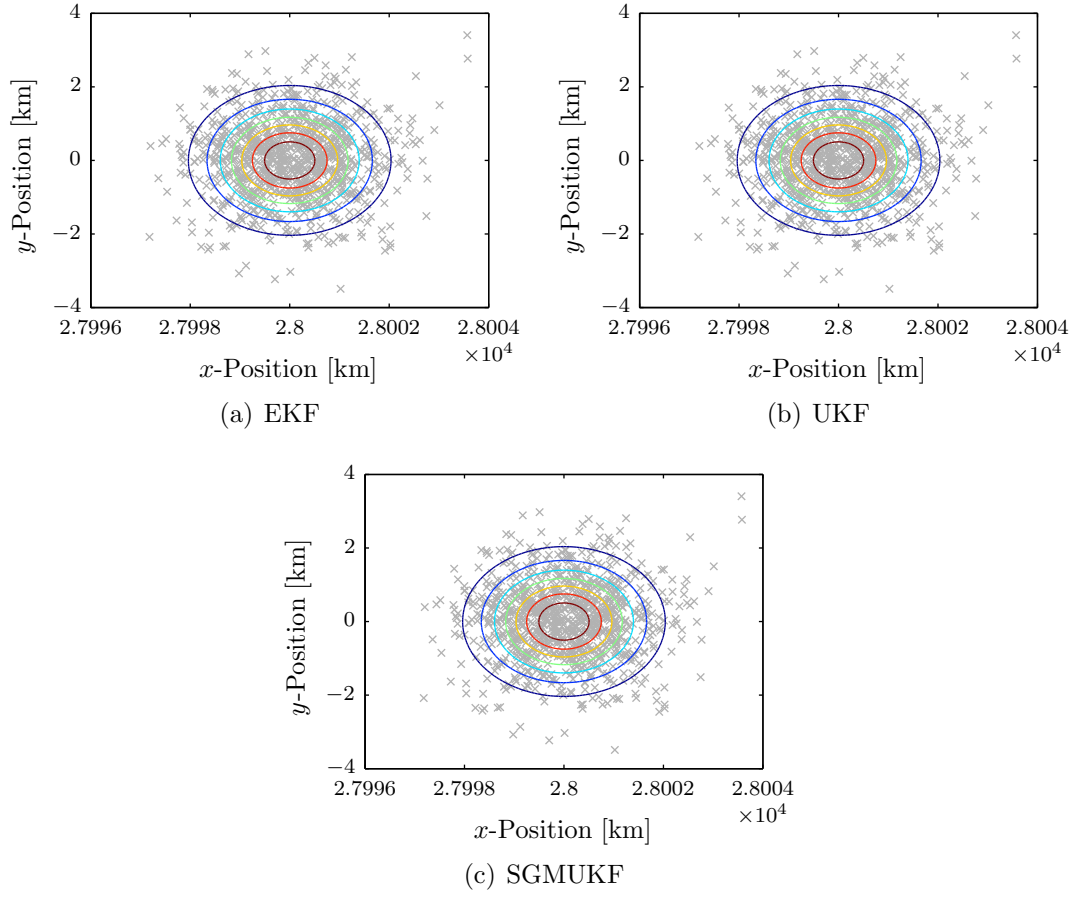


Figure 5.12: Position pdf contours with monte carlo samples at epoch.

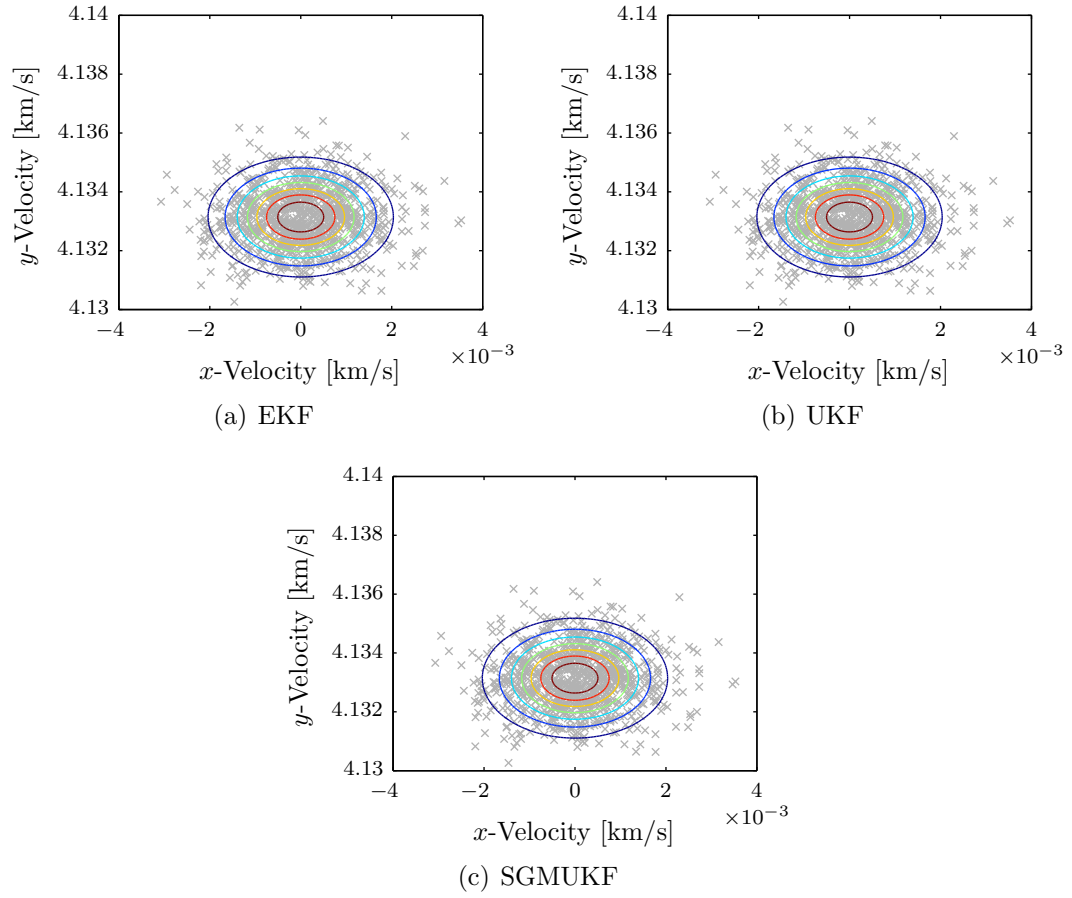


Figure 5.13: Velocity pdf contours with monte carlo samples at epoch.

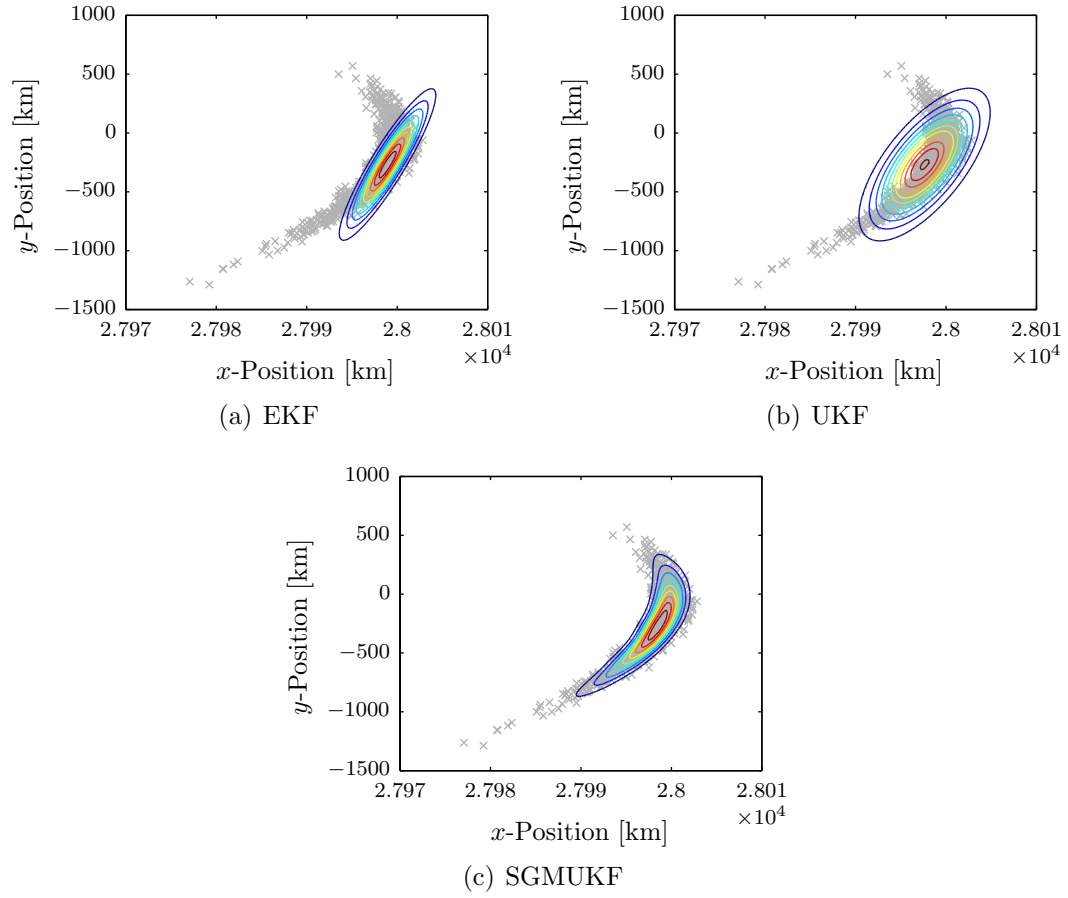


Figure 5.14: Position pdf contours with monte carlo samples at one period of the nominal orbit.

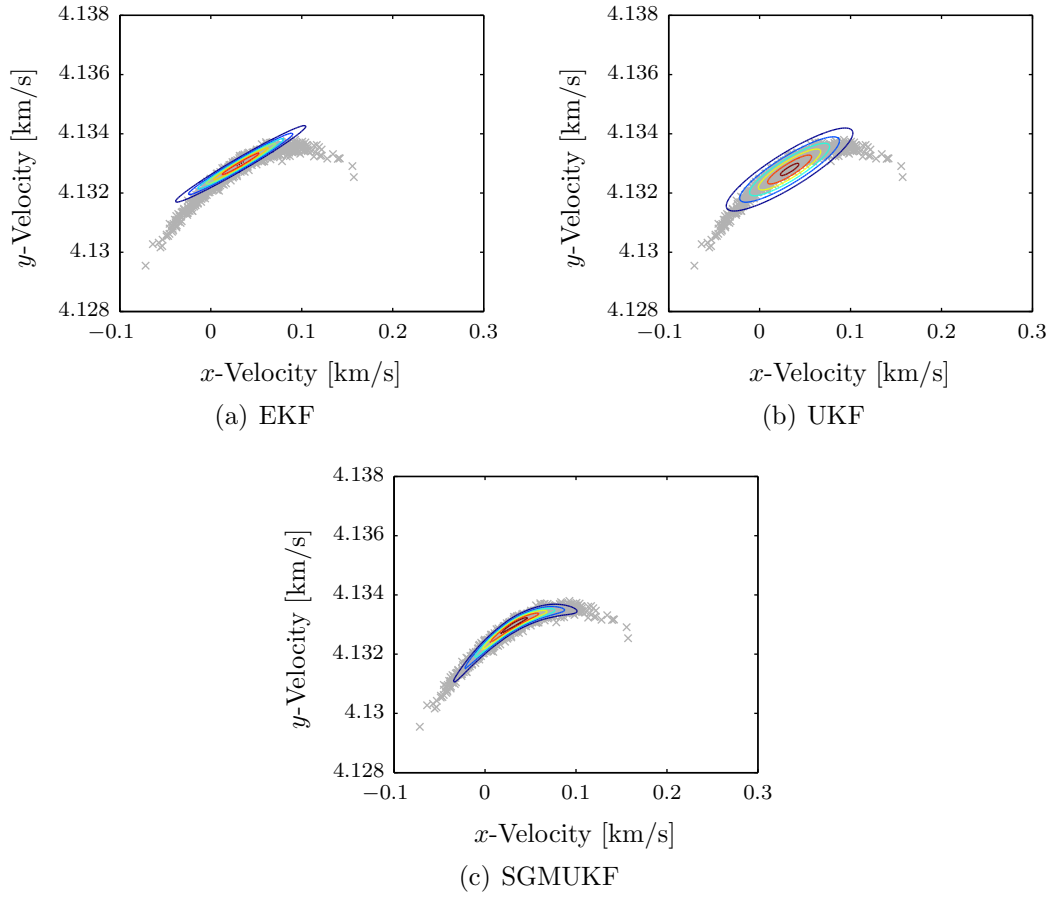


Figure 5.15: Velocity pdf contours with monte carlo samples at one period of the nominal orbit.

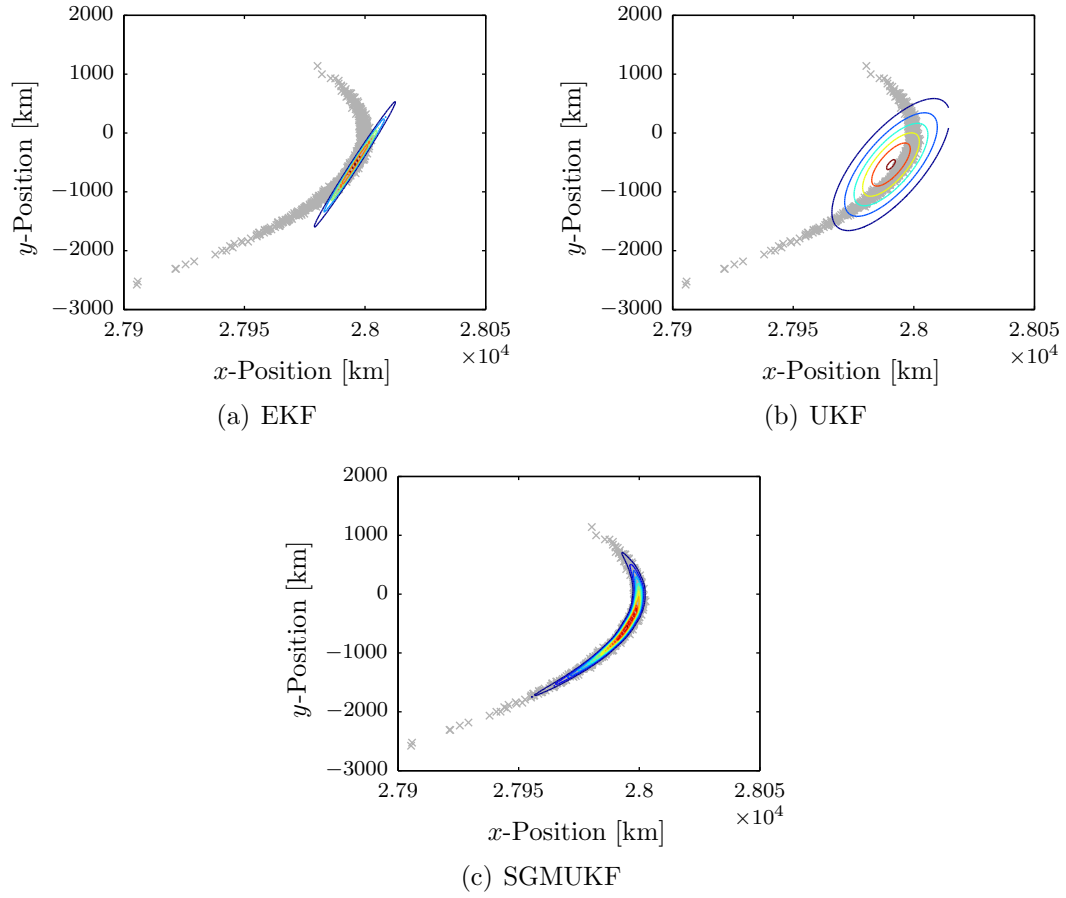


Figure 5.16: Position pdf contours with monte carlo samples at two periods of the nominal orbit.

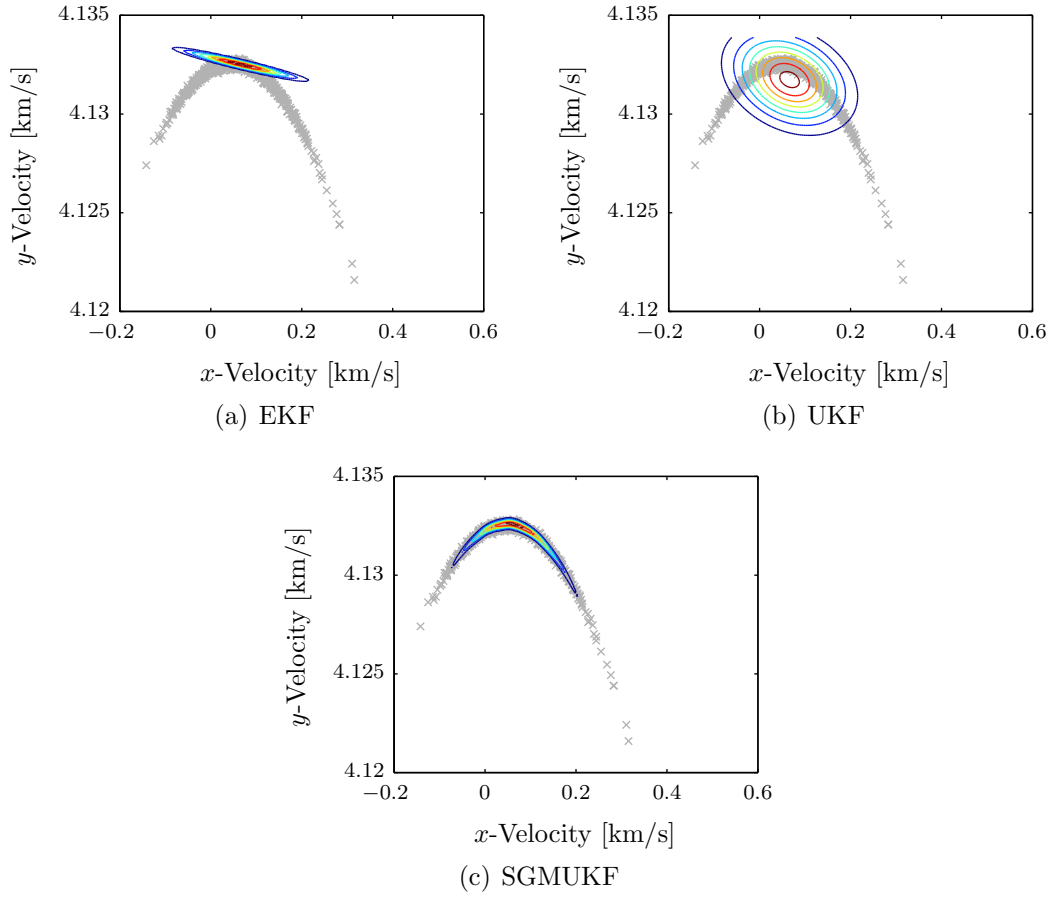


Figure 5.17: Velocity pdf contours with monte carlo samples at two periods of the nominal orbit.



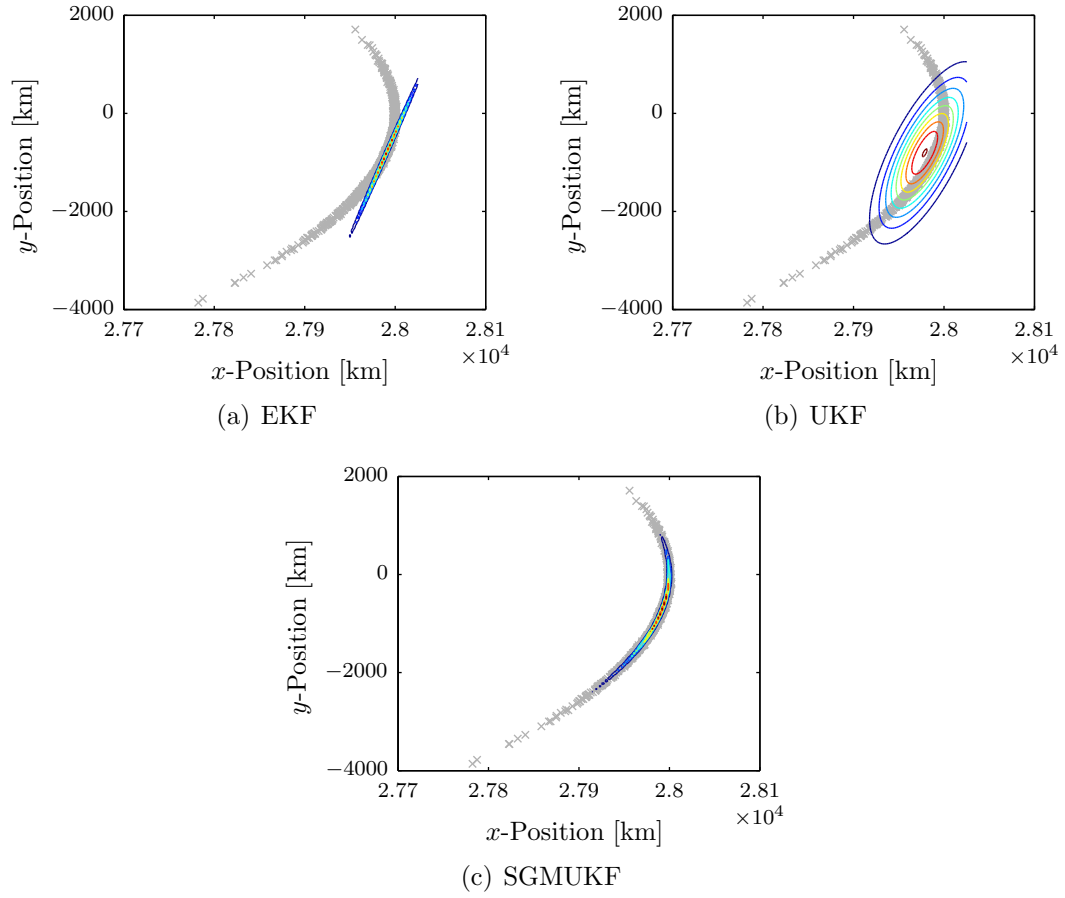


Figure 5.18: Position pdf contours with monte carlo samples at three periods of the nominal orbit.

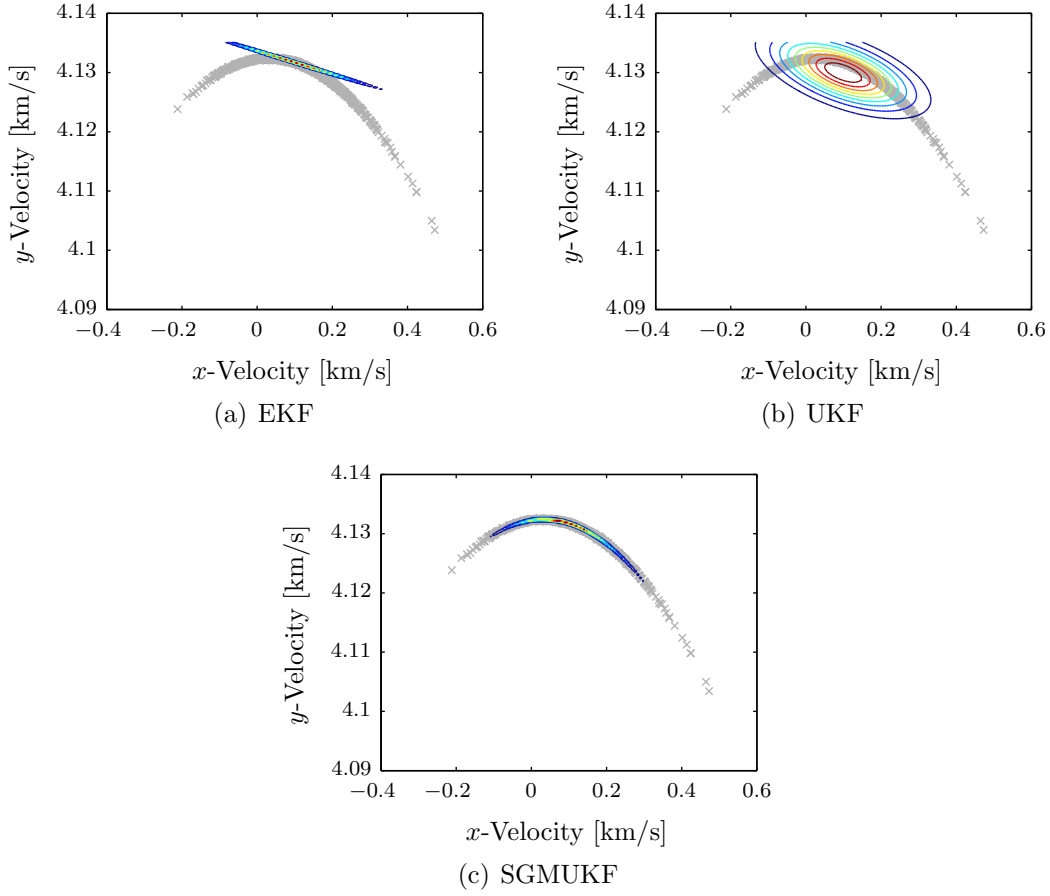


Figure 5.19: Velocity pdf contours with monte carlo samples at three periods of the nominal orbit.

## 5.2 Propagation in the Full Tracking Model

In order to extend the range of testing for the developed methods, a more complex tracking model, termed the full tracking model, is now considered. For the full tracking model, the rotational motion of the vehicle is considered, the active accelerations modeled are that of the central body gravity, third body gravity, and

solar radiation pressure (SRP), and the active moment modeled is that of SRP. Therefore, the full tracking model equations of motion are given by

$$\begin{aligned}\dot{\mathbf{r}}^i &= \mathbf{v}^i \\ \dot{\mathbf{v}}^i &= \mathbf{a}_g^i(\mathbf{r}^i) + \mathbf{a}_{3\text{rd}}^i(\mathbf{r}^i) + \mathbf{a}_{srp}^i(\mathbf{r}^i, \bar{\mathbf{q}}_i^b) \\ \dot{\bar{\mathbf{q}}}_i^b &= \frac{1}{2}\bar{\boldsymbol{\omega}}_{b/i}^b \otimes \bar{\mathbf{q}}_i^b \\ \dot{\boldsymbol{\omega}}_{b/i}^b &= \mathbf{J}^{-1} \left( \mathbf{m}_{srp}^b(\mathbf{r}^i, \bar{\mathbf{q}}_i^b) - \boldsymbol{\omega}_{b/i}^b \times \mathbf{J} \boldsymbol{\omega}_{b/i}^b \right),\end{aligned}$$

Defining the state vector of the nonlinear dynamical system to be the combination of the position, velocity, attitude, and angular velocity of the SRP, the state vector and equations of motion which describe the nonlinear dynamical system may be written as

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{r}^i \\ \mathbf{v}^i \\ \bar{\mathbf{q}}_i^b \\ \boldsymbol{\omega}_{b/i}^b \end{bmatrix} \quad \text{and} \quad \mathbf{f}(\mathbf{x}(t), t) = \begin{bmatrix} \mathbf{v}^i \\ \mathbf{a}_g^i(\mathbf{r}^i) + \mathbf{a}_{3\text{rd}}^i(\mathbf{r}^i) + \mathbf{a}_{srp}^i(\mathbf{r}^i, \bar{\mathbf{q}}_i^b) \\ \frac{1}{2}\bar{\boldsymbol{\omega}}_{b/i}^b \otimes \bar{\mathbf{q}}_i^b \\ \mathbf{J}^{-1} \left( \mathbf{m}_{srp}^b(\mathbf{r}^i, \bar{\mathbf{q}}_i^b) - \boldsymbol{\omega}_{b/i}^b \times \mathbf{J} \boldsymbol{\omega}_{b/i}^b \right) \end{bmatrix}.$$

As was shown in Chapter 4, the linearized dynamics Jacobian for the full tracking model is given by

$$\mathbf{F}(\hat{\mathbf{x}}(t), t) = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_g + \mathbf{A}_{3\text{rd}} + \mathbf{A}_{r,srp} & \mathbf{0} & \mathbf{A}_{\theta,srp} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -[\hat{\boldsymbol{\omega}}_{b/i}^b \times] & \mathbf{I} \\ \mathbf{J}^{-1} \mathbf{M}_{r,srp} & \mathbf{0} & \mathbf{J}^{-1} \mathbf{M}_{\theta,srp} & \mathbf{J}^{-1} \left[ [\mathbf{J} \hat{\boldsymbol{\omega}}_{b/i}^b \times] - [\hat{\boldsymbol{\omega}}_{b/i}^b \times] \mathbf{J} \right] \end{bmatrix}. \quad (5.2)$$

Since the SGMUKF recursive filtering scheme relies on the implementation of either differential entropy or Rényi entropy, it is worthwhile to show that the trace of the linearized dynamics Jacobian for the full tracking model is zero. From Eq. (5.2), it

follows that

$$\text{trace} \{ \mathbf{F}(\hat{\mathbf{x}}(t), t) \} = \text{trace} \{ -[\hat{\boldsymbol{\omega}}_{b/i}^b \times] \} + \text{trace} \{ \mathbf{J}^{-1} [ [\mathbf{J} \hat{\boldsymbol{\omega}}_{b/i}^b \times] - [\hat{\boldsymbol{\omega}}_{b/i}^b \times] \mathbf{J} ] \} . \quad (5.3)$$

From the fact that  $[\hat{\boldsymbol{\omega}}_{b/i}^b \times]$  is a skew-symmetric matrix by definition, the first term in Eq. (5.3) is zero. Therefore, we need only consider the second term of Eq. (5.3), which is given by

$$\begin{aligned} \text{trace} \{ \mathbf{F}(\hat{\mathbf{x}}(t), t) \} &= \text{trace} \{ \mathbf{J}^{-1} [ [\mathbf{J} \hat{\boldsymbol{\omega}}_{b/i}^b \times] - [\hat{\boldsymbol{\omega}}_{b/i}^b \times] \mathbf{J} ] \} \\ &= \text{trace} \{ \mathbf{J}^{-1} [\mathbf{J} \hat{\boldsymbol{\omega}}_{b/i}^b \times] \} - \text{trace} \{ \mathbf{J}^{-1} [\hat{\boldsymbol{\omega}}_{b/i}^b \times] \mathbf{J} \} \end{aligned} \quad (5.4)$$

$$= \text{trace} \{ \mathbf{J}^{-1} [\mathbf{J} \hat{\boldsymbol{\omega}}_{b/i}^b \times] \} - \text{trace} \{ [\hat{\boldsymbol{\omega}}_{b/i}^b \times] \} , \quad (5.5)$$

where the invariance under cyclic permutation property of the trace operator has been used to eliminate  $\mathbf{J}$  and  $\mathbf{J}^{-1}$  in the second term of Eq. (5.4). Again, since  $[\hat{\boldsymbol{\omega}}_{b/i}^b \times]$  is a skew-symmetric matrix, the second term in Eq. (5.5) is zero, yielding

$$\text{trace} \{ \mathbf{F}(\hat{\mathbf{x}}(t), t) \} = \text{trace} \{ \mathbf{J}^{-1} [\mathbf{J} \hat{\boldsymbol{\omega}}_{b/i}^b \times] \} . \quad (5.6)$$

For any  $\mathbf{A} \in \mathbb{R}^{3 \times 3}$  and  $\mathbf{b} \in \mathbb{R}^3$ , with  $A_{i,j}$  representing the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column of matrix  $\mathbf{A}$ , it is readily observed that

$$\text{trace} \{ \mathbf{A}[\mathbf{b} \times] \} = (A_{2,3} - A_{3,2})b_1 + (A_{3,1} - A_{1,3})b_2 + (A_{1,2} - A_{2,1})b_3 .$$

Therefore, for any symmetric  $\mathbf{A}$ , that is  $\mathbf{A} = \mathbf{A}^T$ , it is seen that

$$\text{trace} \{ \mathbf{A}[\mathbf{b} \times] \} = 0 . \quad (5.7)$$

Since the moment of inertia matrix,  $\mathbf{J}$  is symmetric, so then is its inverse, and since Eq. (5.7) is of the same form as the right-hand side of Eq. (5.6), it immediately follows

that

$$\text{trace} \{ \mathbf{J}^{-1} [\mathbf{J} \hat{\boldsymbol{\omega}}_{b/i}^b \times] \} = 0 ,$$

which yields the desired result that the linearized dynamics Jacobian for the full tracking model has zero trace, i.e.

$$\text{trace} \{ \mathbf{F}(\hat{\mathbf{x}}(t), t) \} = 0 .$$

This means that the differential entropy and the Rényi entropy are constant for the linearized dynamical system, which simplifies the implementation of the SGMUKF by allowing the predicted entropy to be compared against some reference value without needing to implement a differential equation to solve for the entropy of the linearized system.

The orbit considered for testing the SGMUKF method is described by the Keplerian elements

$$a = 42165.91 \text{ km} , \quad e = 0.0002429 , \quad i = 0.83^\circ , \quad \Omega = 0^\circ , \quad \omega = 0^\circ , \quad M = 0^\circ ,$$

and is shown in Figure 5.20.

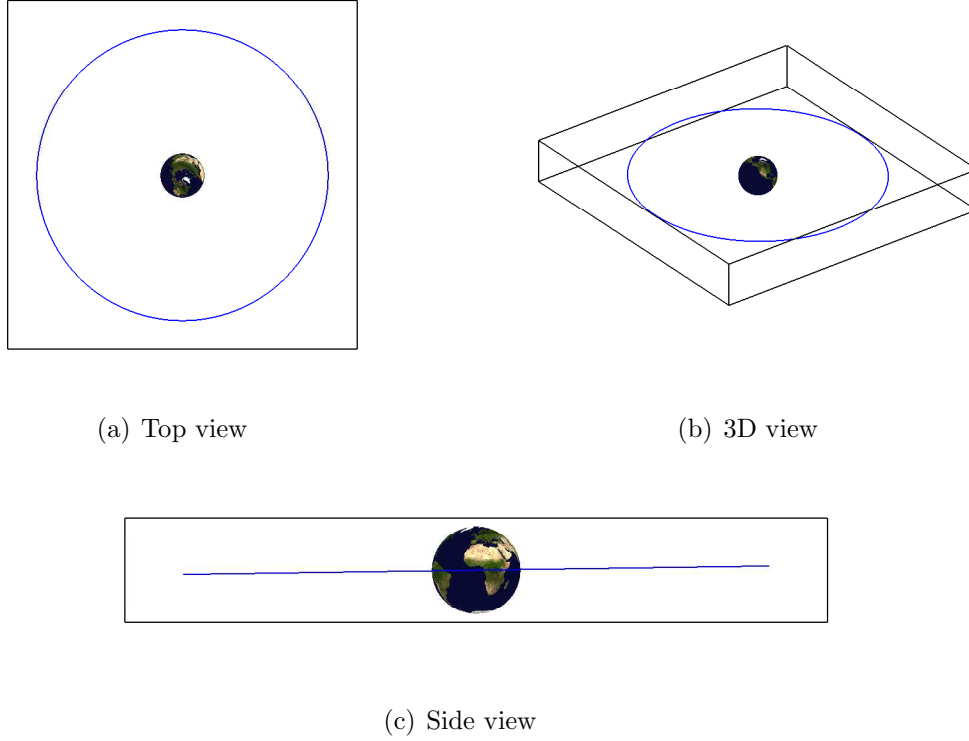


Figure 5.20: Nominal trajectory for the full tracking model.

Furthermore, the initial uncertainty on the position is taken to be 1 [km], the initial uncertainty on the velocity is taken to be 1 [m/s], the initial attitude uncertainty is taken to be  $1^\circ$ , and the initial angular velocity uncertainty is taken to be 0.1 [deg/hr]. The computation of the SRP acceleration and moment depends on implementation of a specific flat plate model for the object, which for the scenario under consideration is described by a hexagonal prism (developed by Rose<sup>54</sup>) as shown in Figure 5.21. This is an 8-plate model with the body-frame unit vectors defined by the unit vector triad  $\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$ . Additionally, the plate normal, denoted for the  $k^{\text{th}}$

plate by  $\mathbf{u}_{n,k}^b$ , is depicted in Figure 5.21. The area,  $A_k$ , and position from the object center,  $\mathbf{r}_{p,k}^b$ , of each plate are fully determined by specifying the side length,  $a$ , and the prism height,  $h$ . These were chosen so as to represent a typical spacecraft bus size, and are taken to be  $a = 2$  [m] and  $h = 4$  [m]. The total object mass was again chosen to be representative of a typical spacecraft mass and is given by  $m = 2688.7$  [kg]. Based upon the mass, side length, and prism height, the moment of inertia can be found to be a diagonal matrix of the form

$$\mathbf{J} = \begin{bmatrix} J_{1,1} & 0 & 0 \\ 0 & J_{2,2} & 0 \\ 0 & 0 & J_{3,3} \end{bmatrix},$$

where the elements of  $\mathbf{J}$  are given by

$$\begin{aligned} J_{1,1} &= m \left( \frac{a^2}{6} + \frac{d^2}{3} + \frac{h^2}{12} \right) \\ J_{2,2} &= m \left( \frac{a^2}{6} + \frac{d^2}{3} + \frac{h^2}{12} \right) \\ J_{3,3} &= m \left( \frac{a^2}{6} + \frac{d^2}{3} \right). \end{aligned}$$

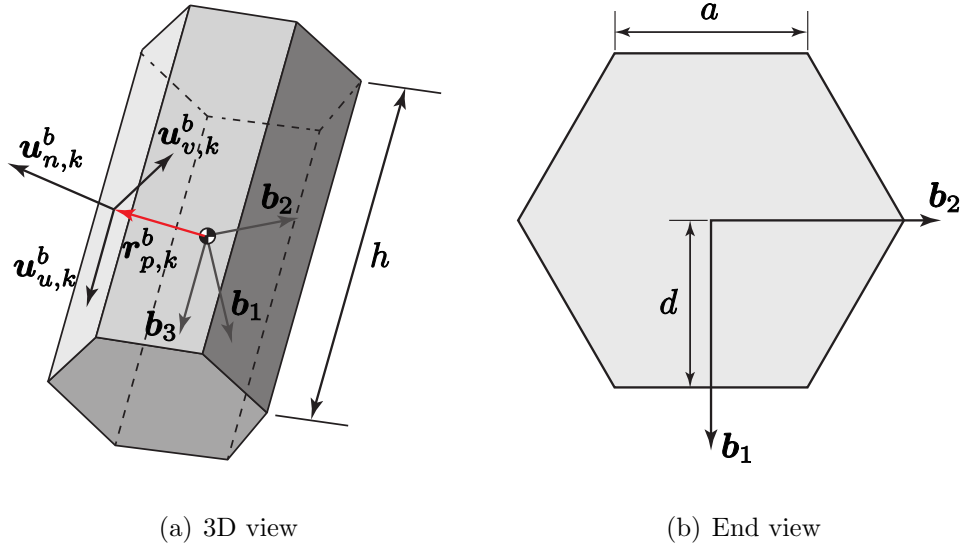


Figure 5.21: Hexagonal prism flat plate model, adapted from [54].

The specification of the diffuse reflectivity and specular reflectivity values for each plate to be used in the SRP acceleration and moment calculations are taken from a TDRS-05 macro plate model described by Lyon.<sup>40</sup> These values, along with the area equations for each plate are given in Table 5.2. The plates in Table 5.2 are specified by the direction in which they are in the body-frame from the center of the object, with  $+x$  denoting the plate that lies along the positive  $\mathbf{b}_1$ ,  $+x, -y$  denoting the plate that lies along the positive  $\mathbf{b}_1$  and negative  $\mathbf{b}_2$  axis, and so on.



Table 5.2: Plate Property Specification for the Hexagonal Prism Model

Plate	Area	Diffuse Reflectivity	Specular Reflectivity
$+x$	$ah$	0.19	0.34
$+x, +y$	$ah$	0.20	0.24
$-x, +y$	$ah$	0.20	0.21
$-x$	$ah$	0.20	0.23
$-x, -y$	$ah$	0.18	0.45
$+x, -y$	$ah$	0.20	0.22
$+z$	$3\sqrt{3}a^2/2$	0.29	0.08
$-z$	$3\sqrt{3}a^2/2$	0.21	0.05

In addition to the SRP acceleration and moments, an  $8 \times 8$  subset of the GGM03C gravity model<sup>64</sup> is implemented, and third body perturbations due to the Sun and Moon are included.

The EKF, UKF, and SGMUKF recursive filtering strategies are then applied to the problem of propagating the initial uncertainty forward in time for one period of the nominal orbit. In applying the SGMUKF methodology, two implementations are considered: one of the implementations uses the 3-component splitting library in the splitting process and the other implementation uses the 5-component splitting library in the splitting process. These are referred to as the 3-component SGMUKF and 5-component SGMUKF, respectively.

To provide a relative measure of the performance between each of the filters,

the likelihood agreement measure is computed using samples from a monte carlo simulation and the predicted pdfs from the EKF, UKF, 3-component SGMUKF, and 5-component SGMUKF. The likelihood measures of the EKF, UKF, and 3-component SGMUKF are normalized by the value for the 5-component SGMUKF so as to provide a relative measure with respect to the 5-component SGMUKF; that is, if the normalized likelihood of the EKF, UKF, or 3-component SGMUKF were to exceed unity, it would be better performing than the 5-component SGMUKF. The covariance for the EKF becomes ill-conditioned with respect to matrix inversion within a short period of time; therefore, its likelihood agreement measure cannot be computed and it is excluded in the plotted results. However, should the EKF not be near-singular, the analysis would be similar to that of the UKF and 3-component SGMUKF. This analysis is summarized in Figure 5.22, wherein it can be observed that the UKF and 3-component SGMUKF are outperformed by the 5-component SGMUKF.

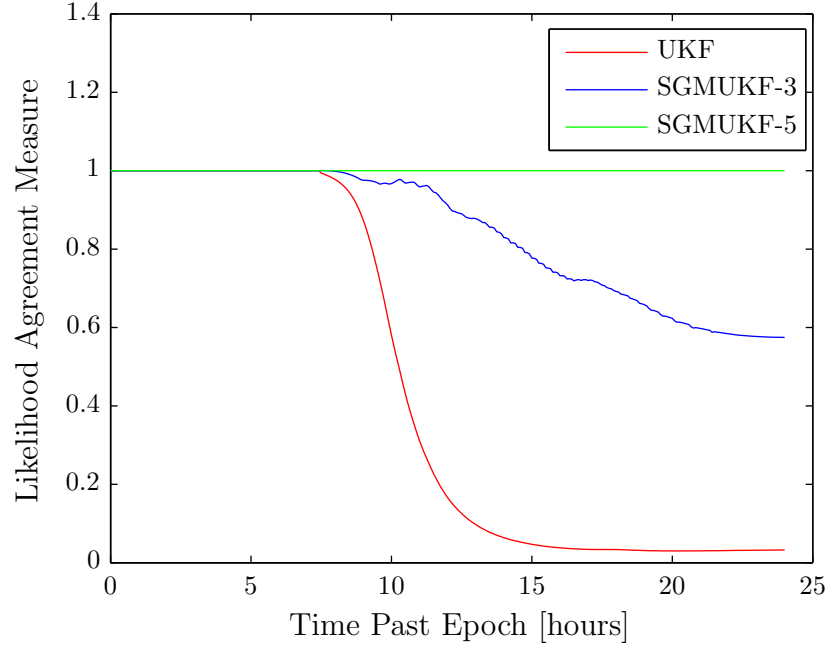


Figure 5.22: Likelihood agreement measure for the UKF, 3-component SGMUKF, and 5-component SGMUKF, normalized by the value for the 5-component SGMUKF

Figures 5.23–5.25 show the position contours at one period of the nominal orbit for each of the planar projections ( $x-y$ ,  $x-z$ , and  $y-z$ ). In each figure, the projected pdf contours are shown for the EKF, the UKF, the 3-component SGMUKF, and the 5-component SGMUKF methods along with samples derived from a monte carlo simulation, which is run by propagating samples drawn from the initial distribution. In both the  $x-y$  and  $x-z$  projections in Figures 5.23 and 5.24, it can be seen that both of the implemented SGMUKF approaches are able to match the curvature that is exhibited by the monte carlo samples. In the  $y-z$  projection, it can be seen that no significant non-Gaussian behavior occurred, yielding the same approximate

results for all of the filtering schemes applied.

Similarly, Figures 5.26–5.28 show the velocity contours, Figures 5.29–5.31 show the attitude contours, and Figures 5.32–5.34 show the angular velocity contours at one period of the nominal orbit for each of the three projections possible. The projected pdf contours are shown for the EKF, the UKF, the 3-component SGMUKF, and the 5-component SGMUKF methods along with samples derived from a monte carlo simulation. The velocity projections show the beginning of non-Gaussian behavior, most specifically in the  $x - y$  projection. Based on the results of Section 5.1, this non-Gaussian behavior would continue growing as the time-scale of the propagation extended. However, in the case of the attitude and angular velocity contours, non-Gaussian behavior has not yet become dominant for this set of initial conditions, yielding very similar performance between the EKF, UKF, and SGMUKF methods with only slight differences between contours for each method.

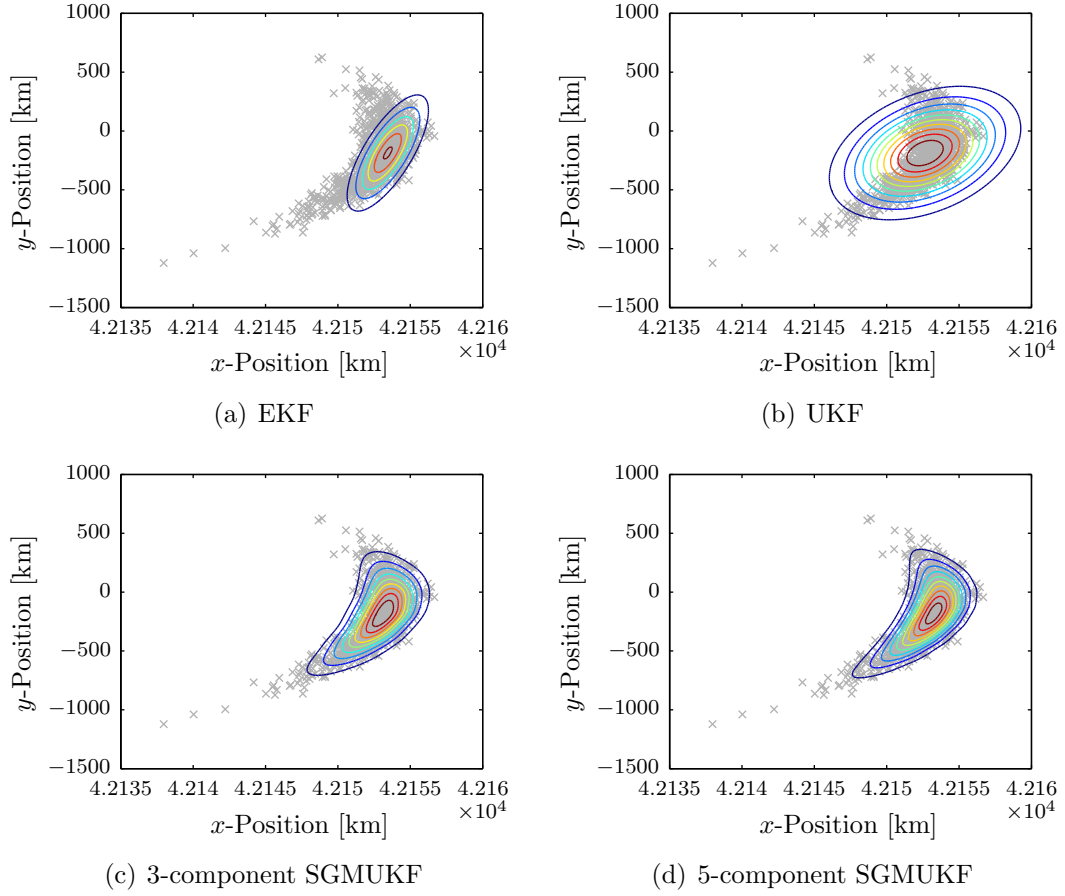
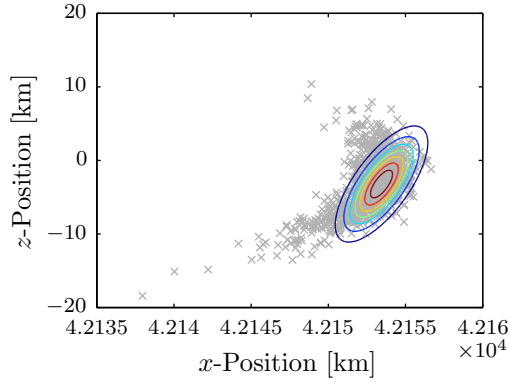
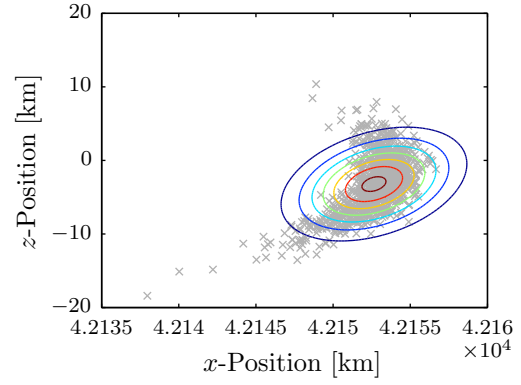


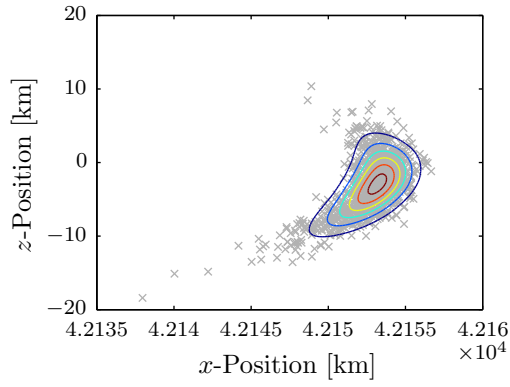
Figure 5.23: Position ( $x - y$  projection) pdf contours with monte carlo samples at one orbit period.



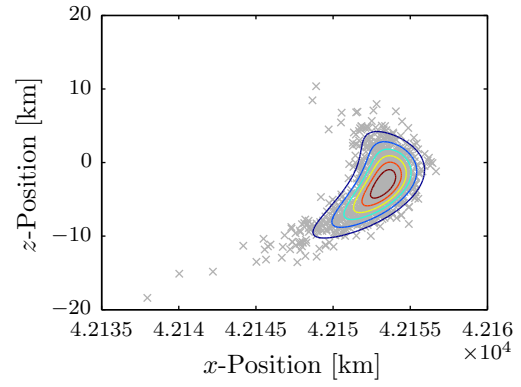
(a) EKF



(b) UKF

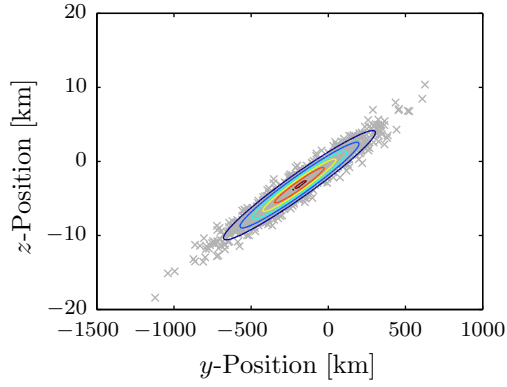


(c) 3-component SGMUKF

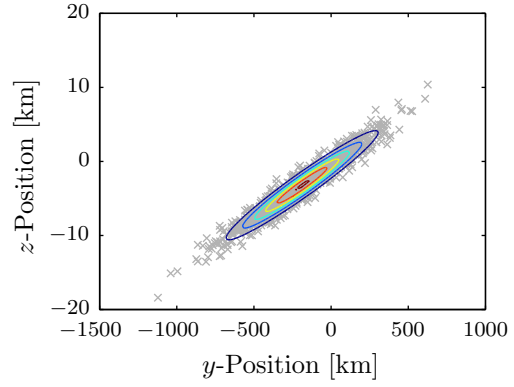


(d) 5-component SGMUKF

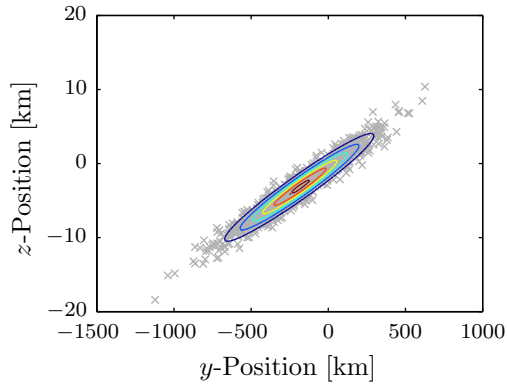
Figure 5.24: Position ( $x - z$  projection) pdf contours with monte carlo samples at one orbit period.



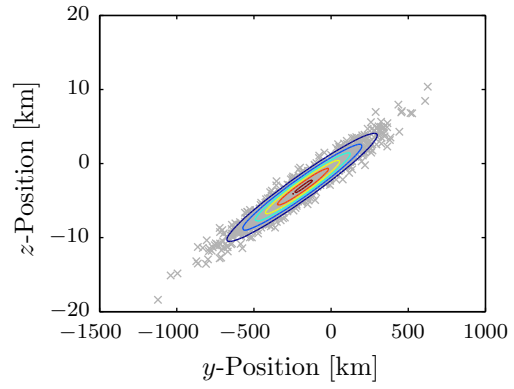
(a) EKF



(b) UKF



(c) 3-component SGMUKF



(d) 5-component SGMUKF

Figure 5.25: Position ( $y-z$  projection) pdf contours with monte carlo samples at one orbit period.

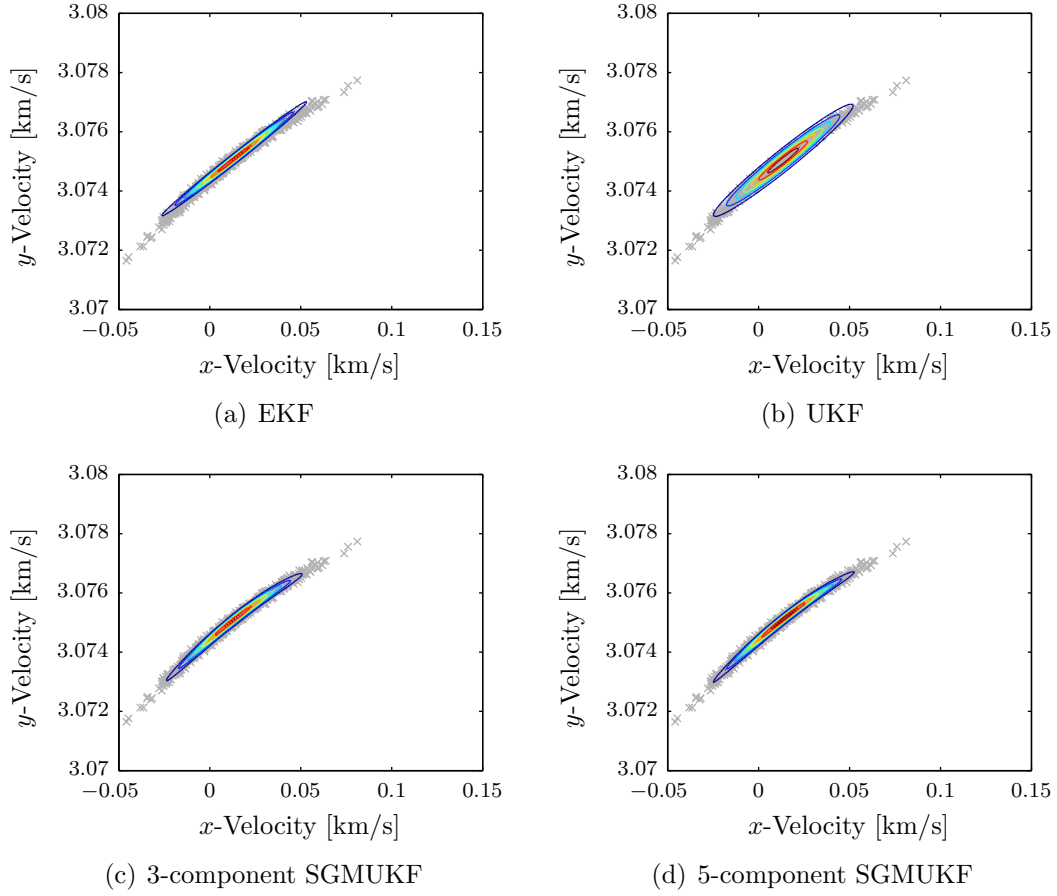


Figure 5.26: Velocity ( $x - y$  projection) pdf contours with monte carlo samples at one orbit period.



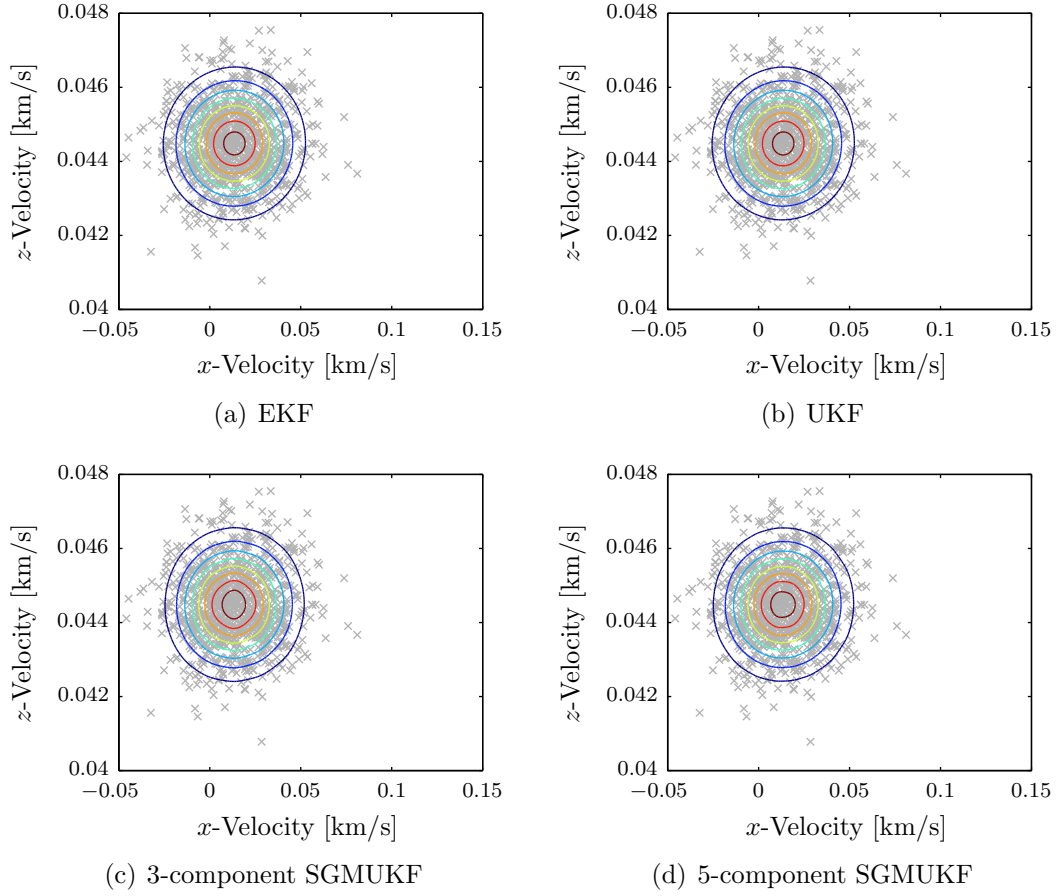


Figure 5.27: Velocity ( $x-z$  projection) pdf contours with monte carlo samples at one orbit period.

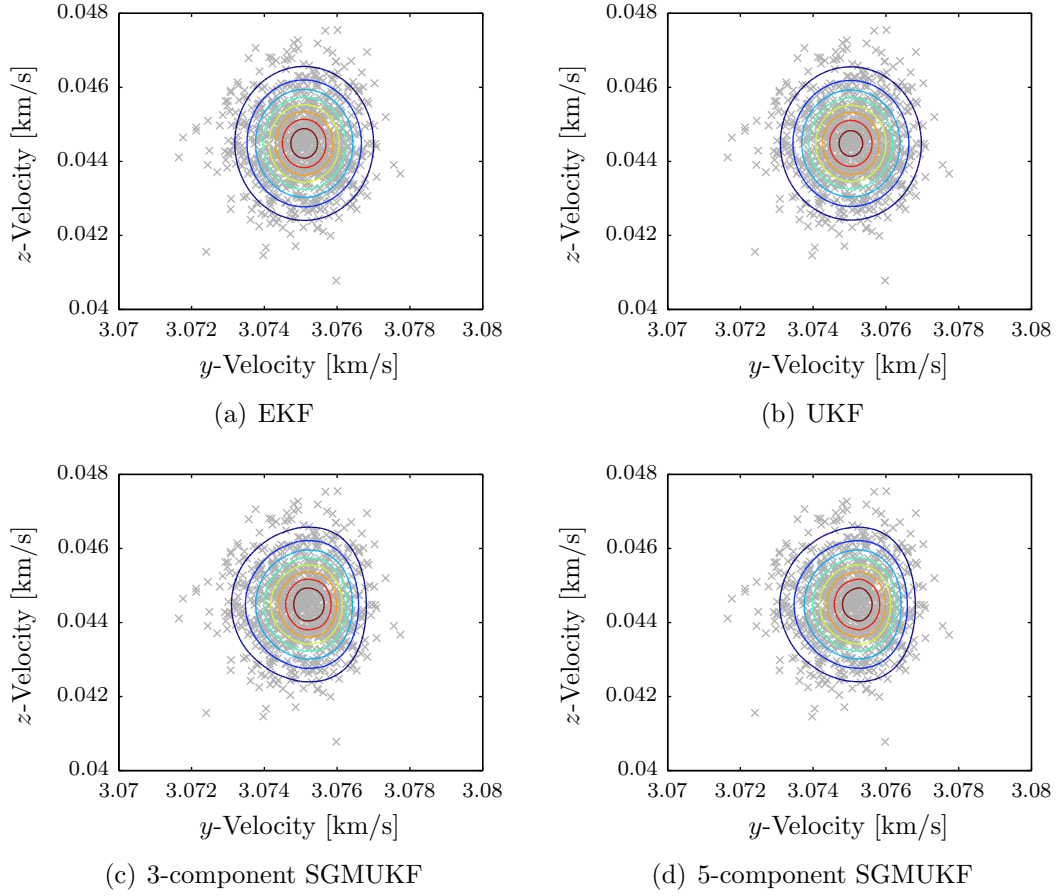
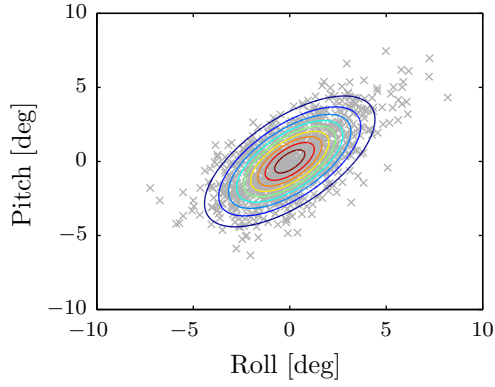
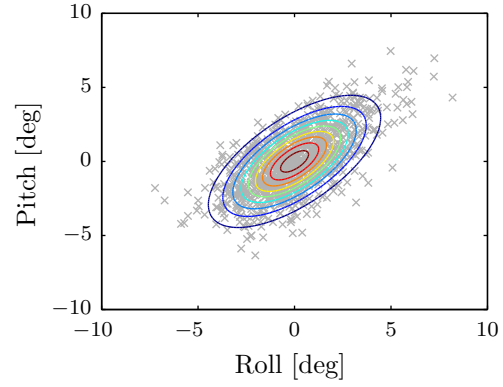


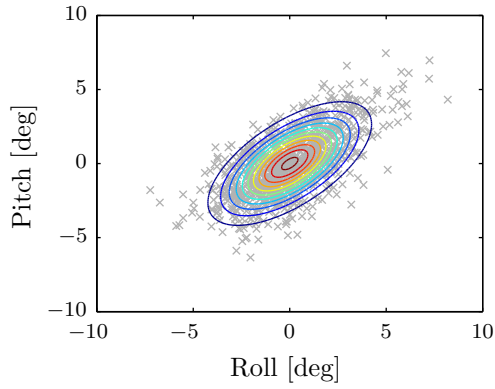
Figure 5.28: Velocity ( $y - z$  projection) pdf contours with monte carlo samples at one orbit period.



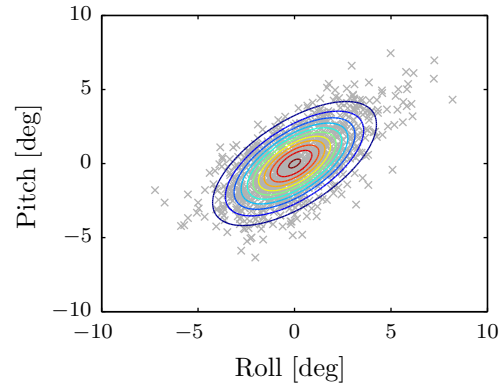
(a) EKF



(b) UKF

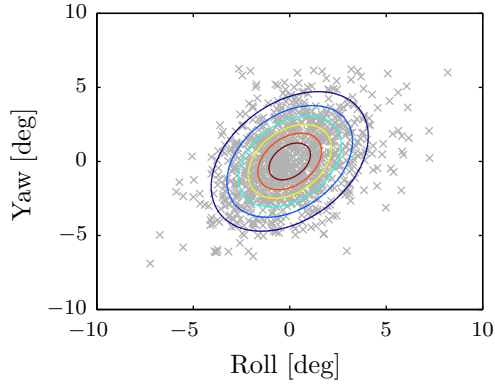


(c) 3-component SGMUKF

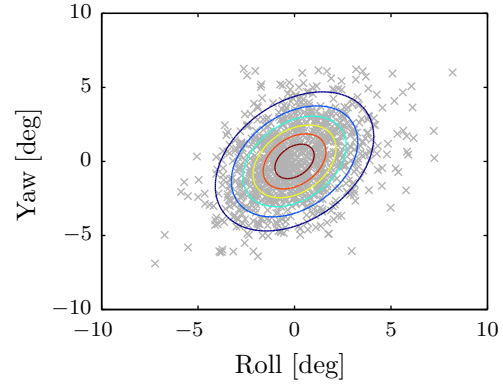


(d) 5-component SGMUKF

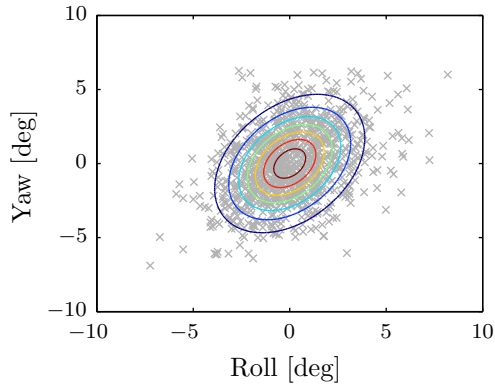
Figure 5.29: Attitude (roll-pitch projection) pdf contours with monte carlo samples at one orbit period.



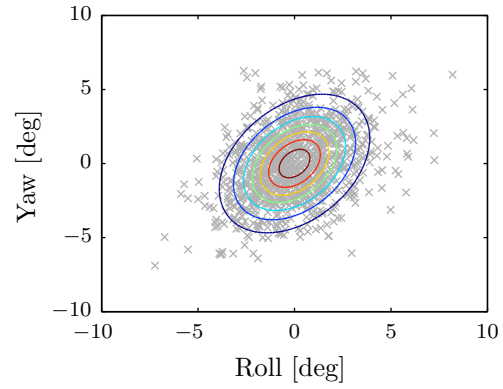
(a) EKF



(b) UKF

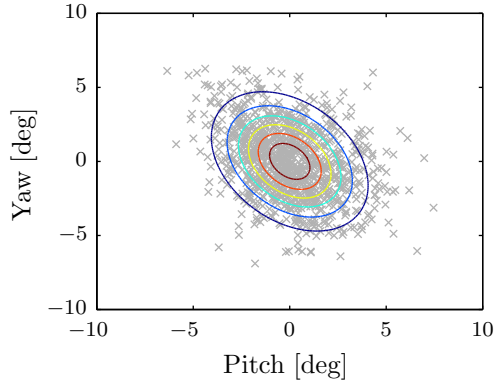


(c) 3-component SGMUKF

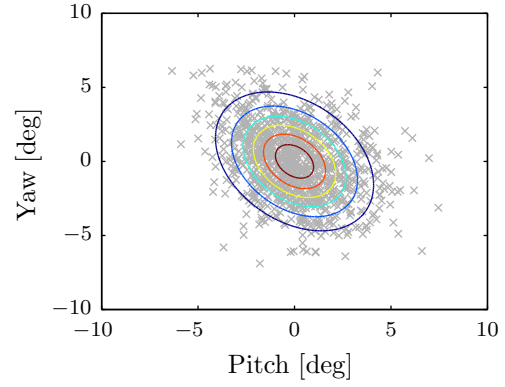


(d) 5-component SGMUKF

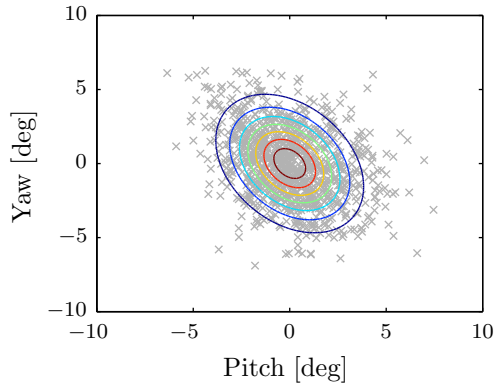
Figure 5.30: Attitude (roll-yaw projection) pdf contours with monte carlo samples at one orbit period.



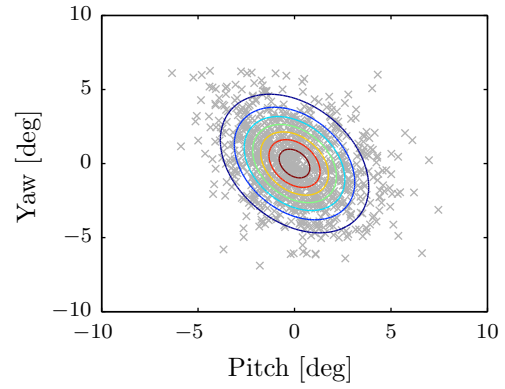
(a) EKF



(b) UKF

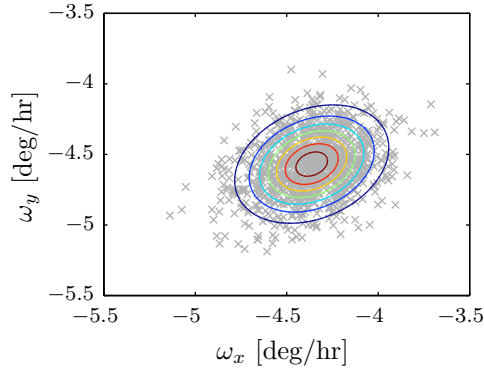


(c) 3-component SGMUKF

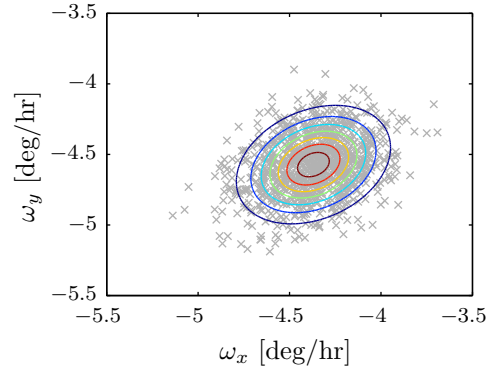


(d) 5-component SGMUKF

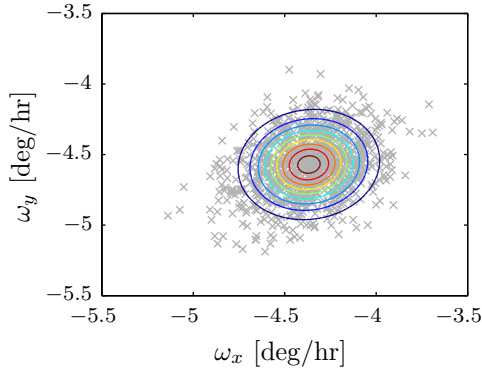
Figure 5.31: Attitude (pitch-yaw projection) pdf contours with monte carlo samples at one orbit period.



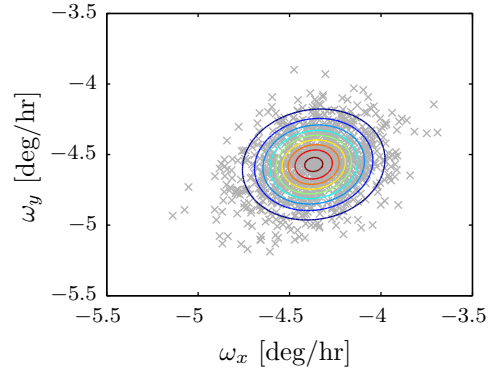
(a) EKF



(b) UKF

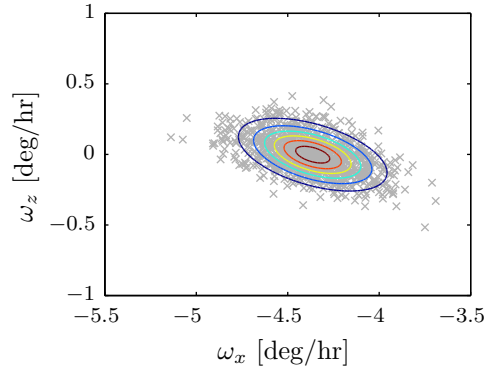


(c) 3-component SGMUKF

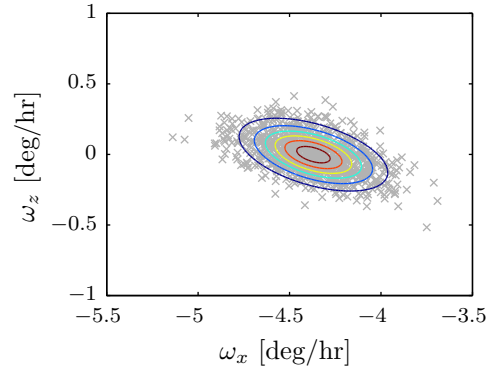


(d) 5-component SGMUKF

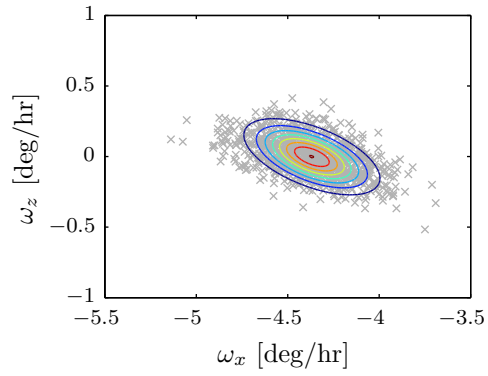
Figure 5.32: Angular Velocity (body-frame  $x - y$  projection) pdf contours with monte carlo samples at one orbit period.



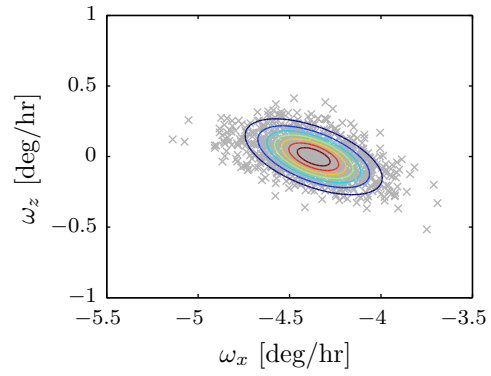
(a) EKF



(b) UKF



(c) 3-component SGMUKF



(d) 5-component SGMUKF

Figure 5.33: Angular Velocity (body-frame  $x - z$  projection) pdf contours with monte carlo samples at one orbit period.

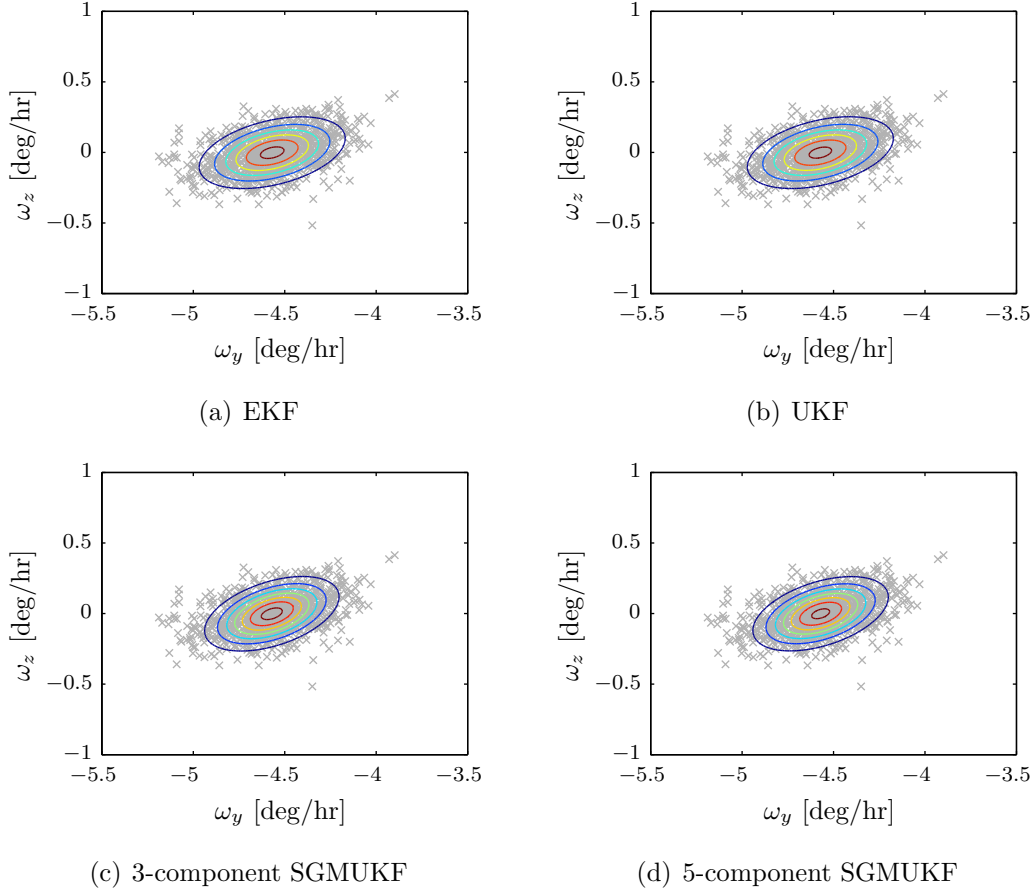


Figure 5.34: Angular Velocity (body-frame  $y-z$  projection) pdf contours with monte carlo samples at one orbit period.

### 5.3 Update in the Full Tracking Model

In order to complete the testing and comparison of the SGMUKF methodology against other methods, we restrict our attention to processing measurement data in the UKF and 5-component SGMUKF which were previously implemented for prop-



agation of uncertainty. The data considered are that of topocentric right ascension and declination angles as detailed in Section 4.2.5. The ground station utilized is given in terms of latitude, longitude, and altitude by

$$\phi = 20.708074^\circ, \quad \lambda = -156.257486^\circ, \quad \text{and} \quad h = 3060.74 \text{ [m]}.$$

To test the efficacy of the SGMUKF and UKF methods, a single sample from the monte carlo run was selected, as shown by the circled point in Figure 5.35. This sample was chosen to be a stressing case for both algorithms in order to address any improvements that may be observed in the SGMUKF with respect to the UKF. From this sample, an arc of 61 measurements of right ascension and declination were generated. The measurements were generated once every 20 seconds for a duration of 20 minutes starting at the same time the propagation phase (as detailed in the previous section) ended. Each of the measurements is subjected to a Gaussian, white-noise sequence with a standard deviation of 1 [arc-second] on both the right ascension and declination angles. The *a priori* pdf at the time of the first measurement, as well as the *a posteriori* pdfs after 1, 2, 10, and 61 measurements were then plotted and are given in Figures 5.36–5.40.

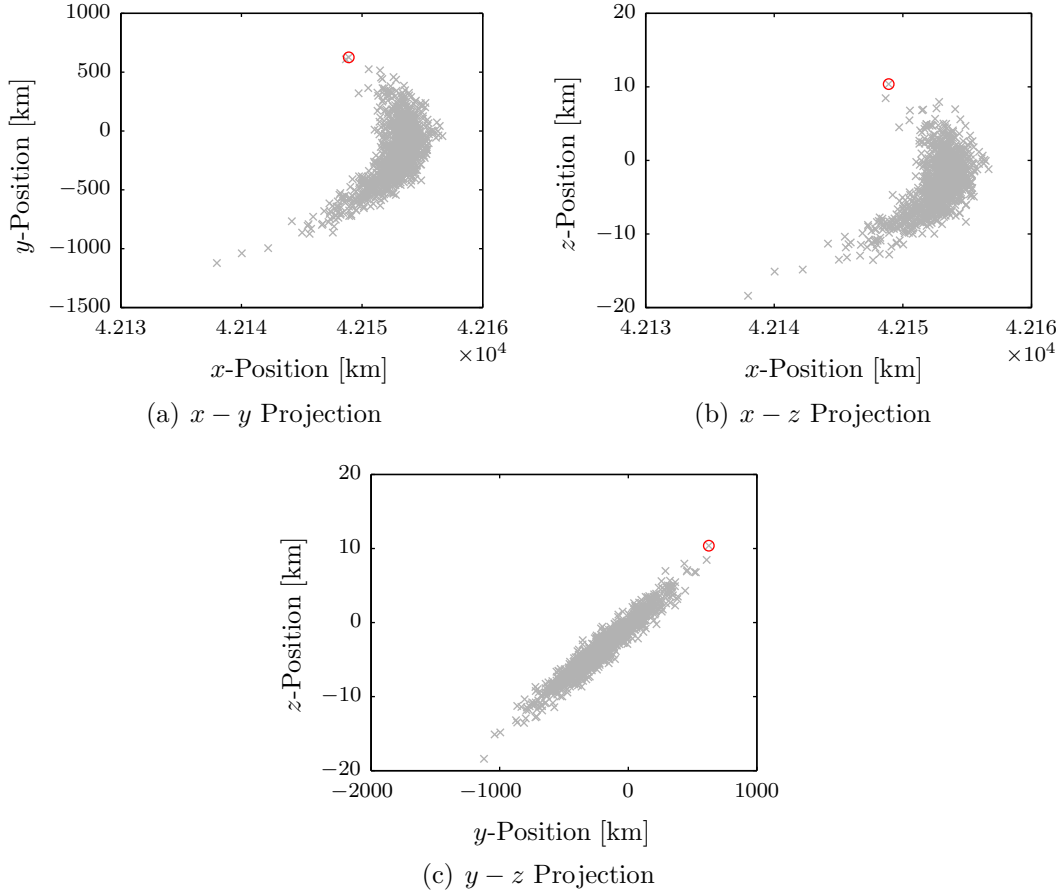


Figure 5.35: Selected point for study of the update in the full tracking model.

To assess the performance of the SGMUKF and UKF methods when processing incoming measurement data, the projected pdf surfaces are plotted in Figures 5.36–5.40. However, instead of the contours of the marginal pdfs, the surfaces of the marginal pdfs are now viewed. Additionally, the true state is plotted in each of the figures to indicate how representative of the true state each pdf is. In order to be able to directly compare the performance of the SGMUKF against that of the UKF,

each projection utilizes the same scale, so that a one-to-one comparison can be made between the marginal pdf that is obtained from the SGMUKF and the one that is obtained from the UKF.

From Figure 5.36 it is seen that the initial pdf indicates a low probability of the true state. However, the SGMUKF predicts curvature of the pdf that leads towards the true state, whereas the UKF pdf does not. Figure 5.37 shows the marginal pdf surfaces and the true state immediately after the first measurement is processed. In each of the SGMUKF marginals, it is observed that after one update the pdf becomes highly Gaussian and no longer retains its curvature. This is to be expected since the measurement pdf is Gaussian and the state pdf is being conditioned upon the measurement pdf. Moreover, since the true state lies on the periphery of the *a priori* distribution, there are very few Gaussian mixture model (GMM) components in the SGMUKF which are surrounding the true state. If the true state had been encompassed by many GMM components, it would have been more likely that a single measurement update would not have led directly to a distinctly Gaussian *a posteriori* state pdf. Also in Figure 5.37 it is observed that the UKF *a posteriori* marginal pdfs are larger than the corresponding SGMUKF ones and less representative of the true state.

After two measurement updates the UKF and SGMUKF both seem to be representing the true state relatively well, as can be seen in Figure 5.38. However, it is also observed that the SGMUKF yields a smaller region of uncertainty than does the UKF. This occurrence is a direct result of the smaller size of the components in the GMM of the SGMUKF. Since the UKF only has one component in its distri-

bution, the single component must represent the entire distribution and is therefore, by necessity, larger. This, in turn, leads to less reduction in the uncertainty when processing measurements as it leads to higher levels of uncertainty in the predicted measurements.

Figure 5.39 shows the UKF and SGMUKF marginal pdfs after ten measurements have been processed. The UKF has become less representative of the true state as illustrated by the true state being near the edge of the UKF distribution. The SGMUKF on the other hand is in excellent agreement with the true state and exhibits a substantially smaller region of uncertainty than does the UKF.

Finally, in Figure 5.40, the marginal pdfs are plotted after all of the measurements from the measurement arc have been processed, a total of 61 measurements. The earlier indications of the superior performance of the SGMUKF to that of the UKF is even more clear as the UKF region of uncertainty still barely retains the true state while the SGMUKF shows good agreement with the truth and a much smaller region of uncertainty.

One byproduct of the measurement conditioning process for the SGMUKF is that it down-weights the components of the GMM distribution which are not in statistical agreement with the measurements. This means that as data are processed, the weights of the components which are not representative of the actual measurement receive successively less weight. By introducing a tolerance on the minimum weight that is retained, the measurement data can be used to prune out the statistically insignificant components of the SGMUKFs GMM model, thereby reducing the number of components implemented in following update or propagation cycles.

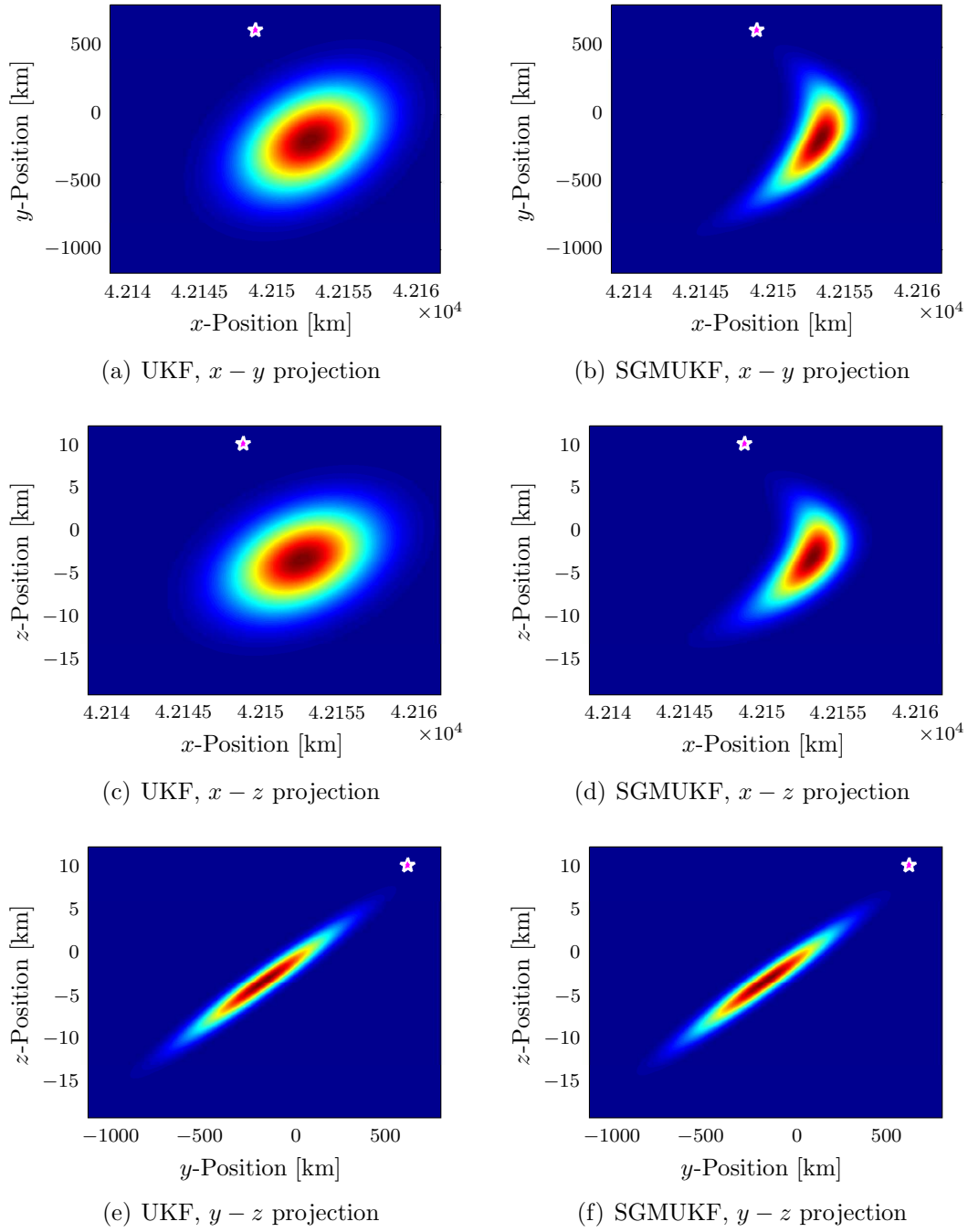


Figure 5.36: Projected position pdf surfaces with true state before any updates.

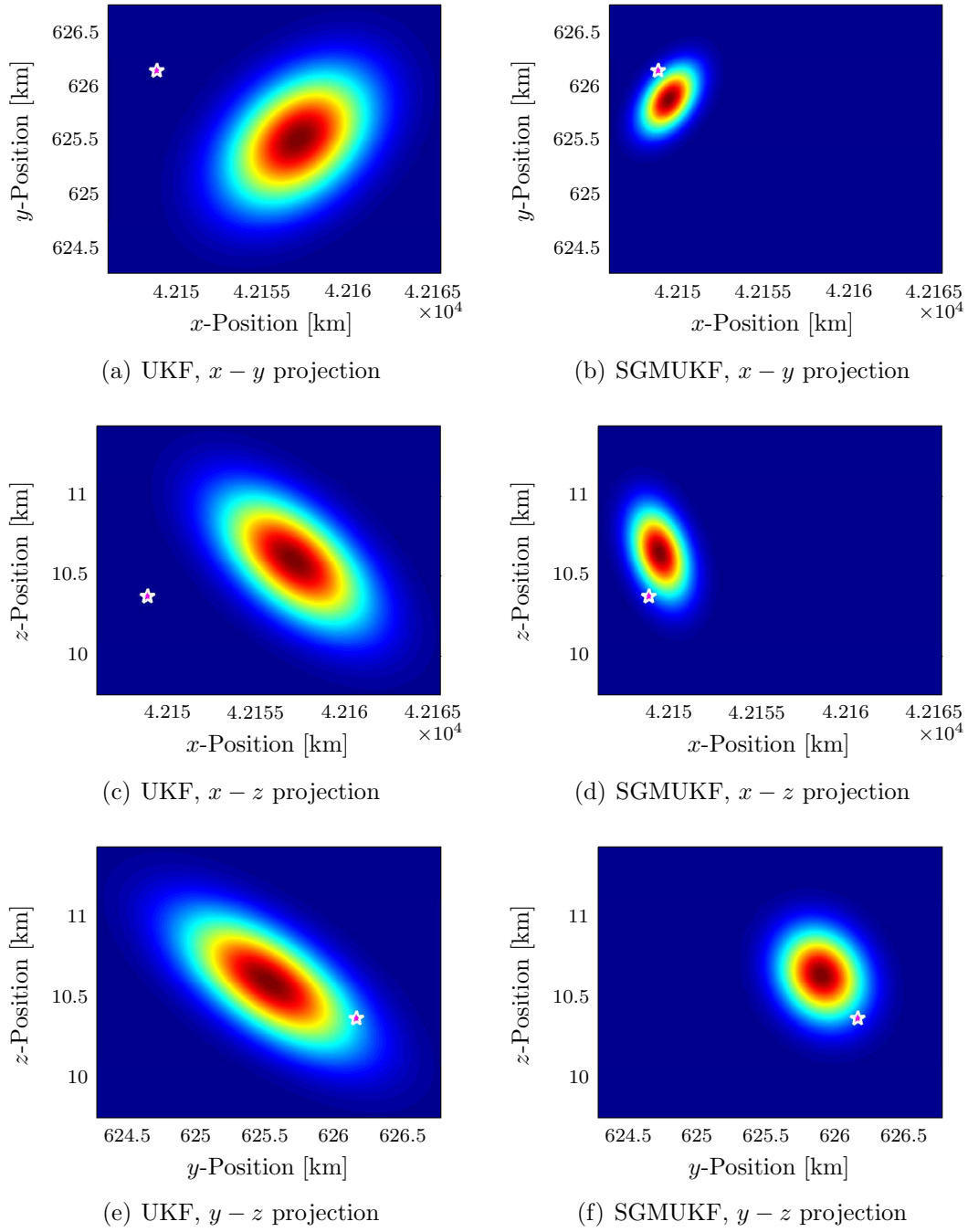


Figure 5.37: Projected position pdf surfaces with true state after one update.

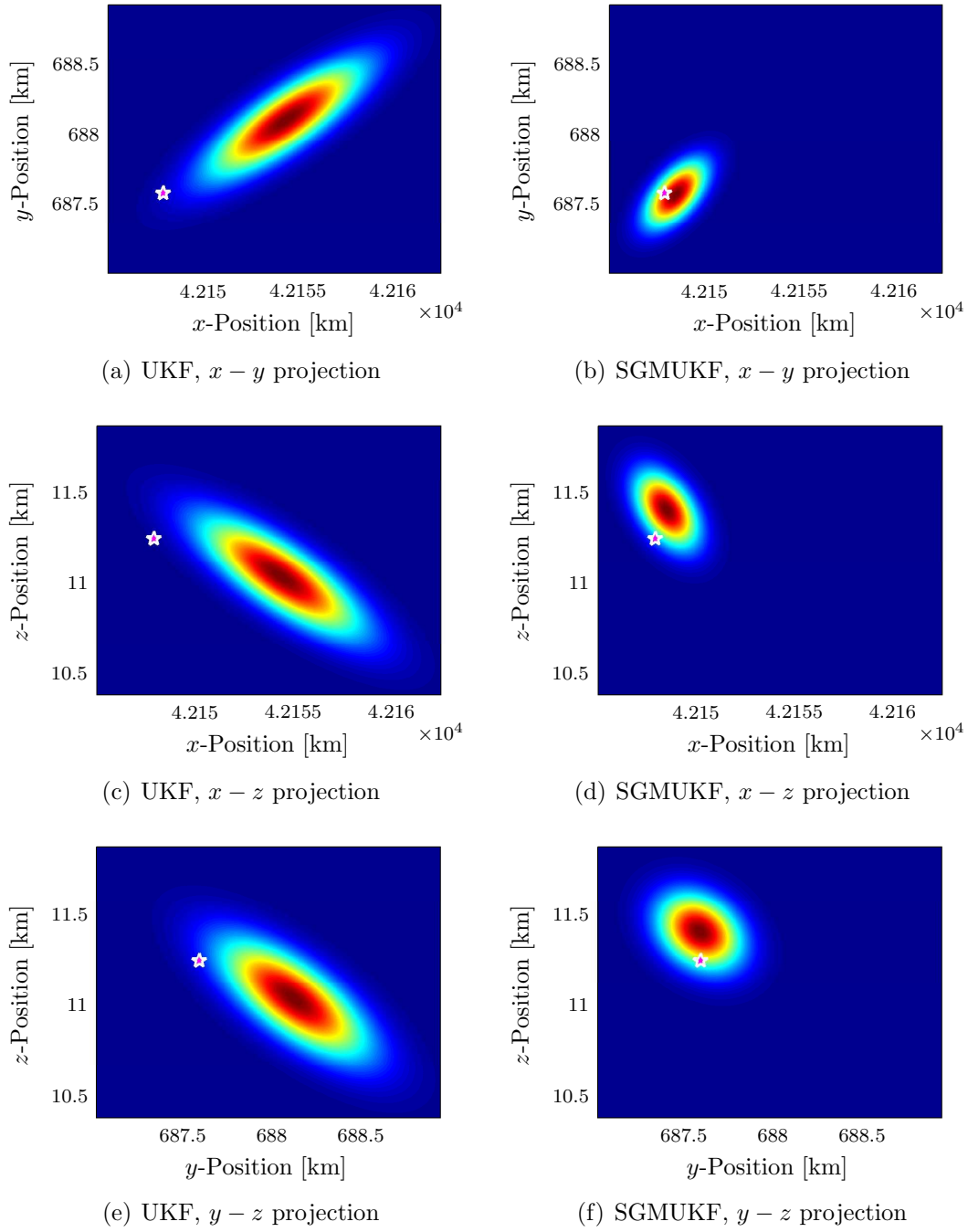


Figure 5.38: Projected position pdf surfaces with true state after two updates.

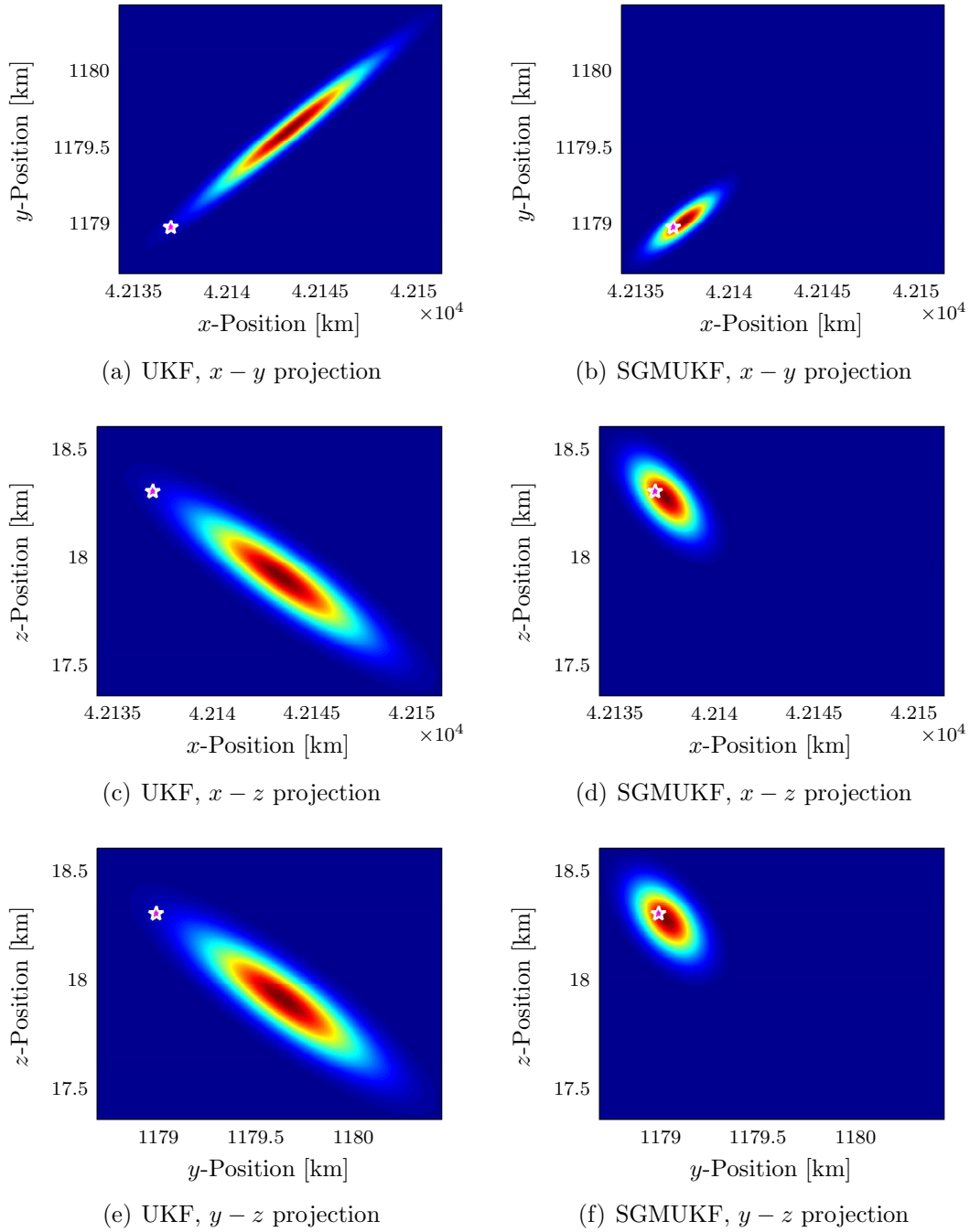


Figure 5.39: Projected position pdf surfaces with true state after ten updates.



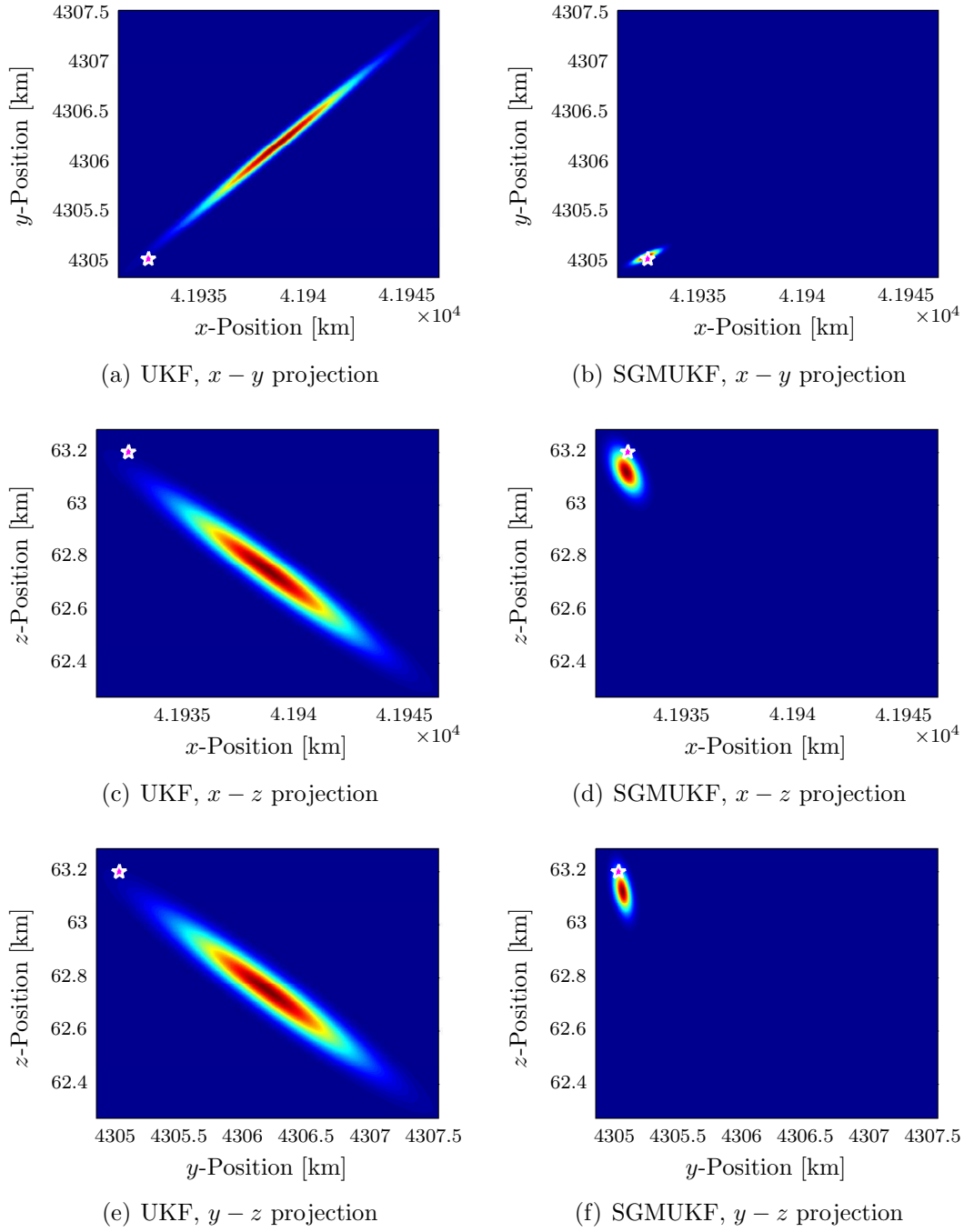


Figure 5.40: Projected position pdf surfaces with true state after all updates.

## Chapter 6

### Conclusions

#### 6.1 Research Summary

The development of a new method, termed the splitting Gaussian mixture unscented Kalman filter (SGMUKF), based on the splitting of Gaussian distributions and the detection of nonlinearity has been presented. The SGMUKF method has been applied to the problem of orbit uncertainty prediction and rectification using low fidelity and high fidelity dynamical models.

Several concepts relating to probability and more specifically operations on probability density functions (pdfs) were presented. Several measures relating to pdfs were detailed, such as the  $L_2$  and  $NL_2$  distances, the likelihood agreement measure to describe the overlap of two pdfs, and the differential and Rényi entropies. Methods for the splitting of univariate and multivariate Gaussian distributions were extended from those available in the literature and subsequently utilized in the application of the SGMUKF.

Models for the simulation of the gravitational acceleration of a central body, the gravitational acceleration due to other celestial bodies, and the acceleration/torques that result from solar radiation pressure (SRP) acting on a macro plate-model of an resident space object (RSO) were presented and utilized in the analysis of the

SGMUKF.

The SGMUKF methodology, along with the more traditional approaches of the extended Kalman filter (EKF) and unscented Kalman filter (UKF), was applied to several problems in orbit uncertainty propagation, including circular orbit and eccentric orbit test cases in a simplified tracking model and a geosynchronous-type orbit in a higher fidelity tracking model. In each test case, the SGMUKF was shown to outperform existing methods. As a final test of the SGMUKF, the rectification of uncertainty via measurement was analyzed and the SGMUKF performance was compared against that of the UKF. Once again, it was shown that the SGMUKF significantly outperformed the existing method.

## 6.2 Future Research Considerations

In developing the SGMUKF approach, the propagation stage of the Gaussian mixture unscented Kalman filter (GMUKF) was modified by detecting nonlinearity during propagation and then applying a multivariate Gaussian distribution splitting algorithm. The update stage, however, was left untouched from that of the GMUKF. While the propagation stage of the SGMUKF ensures that each component remains small enough that linearization can accurately describe the local nonlinear dynamical system, it may be such that the measurement function is still significantly nonlinear with respect to the state variables. To this extent, it would be beneficial to investigate further modifications to the proposed SGMUKF that implement a measurement refinement step to ensure that the *a priori* state pdf is well-represented by linearization over each component.

One benefit of utilizing a Gaussian mixtures approach in general is that it still allows for the conventional understanding of uncertainty volumes that are expressed using mean and covariance, which cannot necessarily be said for a higher-moment algorithm. Additionally, since only the mean and covariance are implemented in the SGMUKF, it is readily adapted to square-root methods, which may allow for increased numerical stability and accuracy in the implementations. The square-root methods can even be used with the differential entropy or Rényi entropy. To see this, recall the Rényi entropy as

$$R_{\kappa}(\mathbf{x}) = \frac{1}{2} \log \left| 2\pi\kappa^{\frac{1}{\kappa-1}} \mathbf{P} \right|.$$

Then, define the square-root factor  $\mathbf{S}$  such that  $\mathbf{P} = \mathbf{S}\mathbf{S}^T$ , such that

$$\begin{aligned} R_{\kappa}(\mathbf{x}) &= \frac{1}{2} \log \left| 2\pi\kappa^{\frac{1}{\kappa-1}} \mathbf{S}\mathbf{S}^T \right| \\ &= \log \left| \sqrt{2\pi\kappa^{\frac{1}{\kappa-1}}} \mathbf{S} \right|. \end{aligned}$$

Only the Rényi entropy was presented here, but the same process holds for the differential entropy. Therefore, given a square-root application, the Rényi entropy or differential entropy can be implemented in the same manner for nonlinearity detection.

## Appendices

# Appendix A

## Attitude Considerations

Unlike representations of translational position in which there are only but a few options, the options for representing the rotational state of an object are quite numerous. The methods commonly employed in this work are that of the rotation matrix and the quaternion. For a more in-depth treatment of attitude representations, see Shuster.<sup>58</sup> The aim herein is to establish the basics of the representations of attitude as well as discuss alterations to some previously presented algorithms when using quaternions.

Since the rotation matrix is commonly utilized in the mapping of vectors from one frame to another, the quaternion is often converted to a rotation matrix in order to accomplish the frame rotation. First of all, a comment on notation is needed. Quaternion are denoted by bold lowercase symbols with overbars. Furthermore, the vector part of the quaternion is given by the same symbol in bold with no overbar and the scalar part of the quaternion is given by the same symbol in non-bold with no overbar. That is, for a quaternion  $\bar{\mathbf{q}}$ , we have

$$\bar{\mathbf{q}} = \begin{bmatrix} \mathbf{q} \\ q \end{bmatrix},$$

where  $\mathbf{q}$  is the vector part of the quaternion and  $q$  is the scalar part of the quaternion.

With this convention, the conversion of a quaternion to a rotation matrix is given by

$$\mathbf{T} = \mathbf{I} - 2q[\mathbf{q} \times] + 2[\mathbf{q} \times]^2. \quad (\text{A.1})$$

Quaternion multiplication (i.e. the composition of rotations) for any two arbitrary quaternions,  $\bar{\mathbf{p}}$  and  $\bar{\mathbf{r}}$  is defined such that multiplication of the respective rotation matrices occurs in the same order, and is given by

$$\bar{\mathbf{p}} \otimes \bar{\mathbf{r}} = \begin{bmatrix} \mathbf{p} \\ p \end{bmatrix} \otimes \begin{bmatrix} \mathbf{r} \\ r \end{bmatrix} = \begin{bmatrix} p\mathbf{r} + r\mathbf{p} - \mathbf{p} \times \mathbf{r} \\ pr - \mathbf{p}^T \mathbf{r} \end{bmatrix},$$

where the symbol  $\otimes$  is used to denote quaternion multiplication. In the special case that the quaternion is used to represent a small rotation, it is such that the small angle quaternion  $\delta\bar{\mathbf{q}}$  is given by

$$\delta\bar{\mathbf{q}} = \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta} \\ 1 \end{bmatrix},$$

where  $\delta\boldsymbol{\theta}$  is a vector of small angles.

## A.1 Averaging Quaternions

In many of the previously presented algorithms (i.e. the unscented transform (UT) and the method of moments), the average value of state variables is required, for instance the weighted average of the sigma-points in the UT, or the weighted average of the means in the method of moments. When the state includes a quaternion, the simple implementation of a normalized weighted sum no longer provides an average that retains the properties of a quaternion. Therefore, an averaging algorithm for quaternions must be implemented. In the cases mentioned, the

problem is to determine an average quaternion given a set of  $n$  quaternions  $\bar{\mathbf{q}}_i$  with associated scalar weights  $w_i$ . As Markley<sup>42</sup> notes, the simple procedure of determining the average quaternion via

$$\bar{\mathbf{q}}_{\text{avg}} = \frac{1}{w_{\text{tot}}} \sum_{i=1}^n \bar{\mathbf{q}}_i \quad \text{where} \quad w_{\text{tot}} = \sum_{i=1}^n w_i \quad (\text{A.2})$$

presents two notable issues. First, the average quaternion is not necessarily unit norm. And second, since  $\bar{\mathbf{q}}_i$  and  $-\bar{\mathbf{q}}_i$  represent the same attitude, an averaging algorithm should not be susceptible to sign changes in  $\bar{\mathbf{q}}_i$ , which Eq. (A.2) is. Markley<sup>42</sup> fully addresses the problem of determining a proper average quaternion; therefore, the details of the derivation are omitted and only the final results are given. To determine the average quaternion, first compute a matrix  $\mathbf{M} \in \mathbb{R}^{4 \times 4}$  as

$$\mathbf{M} = \sum_{i=1}^n w_i \bar{\mathbf{q}}_i \bar{\mathbf{q}}_i^T.$$

Then, it can be shown that the average quaternion is given by the maximization problem

$$\bar{\mathbf{q}}_{\text{avg}} = \underset{\bar{\mathbf{q}} \in \mathbb{S}^3}{\text{argmax}} \bar{\mathbf{q}}^T \mathbf{M} \bar{\mathbf{q}}. \quad (\text{A.3})$$

The maximization problem of Eq. (A.3) can then be case in terms of finding  $\bar{\mathbf{q}}$  that maximizes the performance index

$$J = \bar{\mathbf{q}}^T \mathbf{M} \bar{\mathbf{q}}. \quad (\text{A.4})$$

However, the constraint on the norm of the quaternion must be accounted for, which can be accomplished by adjoining the unit norm constraint to the performance index



with a Lagrange multiplier,  $\lambda$ , such that the augmented performance index is given by

$$J' = \bar{\mathbf{q}}^T \mathbf{M} \bar{\mathbf{q}} + \lambda(\bar{\mathbf{q}}^T \bar{\mathbf{q}} - 1).$$

It is then straightforward to show that the first-order optimality conditions lead to an eigenvalue problem of the form<sup>29</sup>

$$\mathbf{M} \bar{\mathbf{q}} = \lambda \bar{\mathbf{q}},$$

and that the performance index of Eq. (A.4) is given by

$$J = \lambda.$$

Therefore, since the goal is to maximize the performance index, the average quaternion is given by the eigenvector of  $\mathbf{M}$  corresponding to the largest eigenvalue of  $\mathbf{M}$ . It is noted then that this procedure not only leads to an average quaternion which is unit norm, but also leads to a process that is not affected by a sign change in any of the  $\bar{\mathbf{q}}_i$  terms since the performance index is in quadratic form.

## A.2 The Gaussian and Gaussian Mixture Model Distributions

Since computation of either the Gaussian probability density function (pdf) or the Gaussian mixture model (GMM) pdf that were presented in Section 2.1 require the subtraction of the random vector from its mean, adaptation of the computation of the Gaussian pdf must be considered when attitude is part of the random variable. Since the GMM pdf encompasses the case of a single Gaussian pdf, only the GMM

pdf will be considered. Recall the GMM pdf from Section 2.1, which was given by a weighted sum of Gaussian pdfs to be

$$p(\mathbf{x}) = \sum_{i=1}^L w_i p_g(\mathbf{x}; \mathbf{m}_i, \mathbf{P}_i), \quad (\text{A.5})$$

where each Gaussian component has the form

$$p_g(\mathbf{x}; \mathbf{m}_i, \mathbf{P}_i) = |2\pi\mathbf{P}_i|^{-1/2} \exp \left\{ -\frac{1}{2}(\mathbf{x} - \mathbf{m}_i)^T \mathbf{P}_i^{-1} (\mathbf{x} - \mathbf{m}_i) \right\}.$$

Furthermore, to retain the properties of a valid pdf (that is, to ensure positivity across the support of the pdf and to ensure that the area under the pdf is one), the weights must all be positive and must sum to one, that is

$$w_i \geq 0 \quad \forall \quad i \in \{1, 2, \dots, L\} \quad \text{and} \quad \sum_{i=1}^L w_i = 1.$$

To adapt the evaluation of the GMM pdf of Eq. (A.5) for use with quaternions, let the random variable  $\mathbf{x}$  and the set of means  $\mathbf{m}_i$  be broken into non-quaternion and quaternion parts, such that

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \bar{\mathbf{x}}_2 \end{bmatrix} \quad \text{and} \quad \mathbf{m}_i = \begin{bmatrix} \mathbf{m}_{1,i} \\ \bar{\mathbf{m}}_{2,i} \end{bmatrix}.$$

Furthermore, define  $\mathbf{d}_i = \mathbf{x} - \mathbf{m}_i$ , and let it be broken into parts as

$$\mathbf{d}_i = \begin{bmatrix} \mathbf{d}_{1,i} \\ \bar{\mathbf{d}}_{2,i} \end{bmatrix}.$$

Since  $\mathbf{d}_{1,i}$  involves subtraction of the non-quaternion parts of  $\mathbf{x}$  and  $\mathbf{m}_{1,i}$ , no alteration needs to be made. However, to compute  $\mathbf{d}_{2,i}$ , consider the rotational difference of  $\bar{\mathbf{x}}_2$  from  $\bar{\mathbf{m}}_{2,i}$  as given by

$$\delta \bar{\mathbf{q}}_i = \bar{\mathbf{x}}_2 \otimes \bar{\mathbf{m}}_{2,i}^{-1}. \quad (\text{A.6})$$

Then, assuming that  $\delta\bar{\mathbf{q}}_i$  can be well represented by a small angle quaternion yields

$$\delta\bar{\mathbf{q}}_i = \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}_i \\ 1 \end{bmatrix},$$

such that the vector of small rotation angles described by  $\delta\boldsymbol{\theta}_i$  can be computed as twice the vector part of  $\delta\bar{\mathbf{q}}_i$ , or

$$\delta\boldsymbol{\theta}_i = 2 \text{vec} \left( \bar{\mathbf{x}}_2 \otimes \bar{\mathbf{m}}_{2,i}^{-1} \right),$$

where  $\text{vec}(\cdot)$  denotes taking only the vector part of the quaternion argument. The distance  $d_{2,i}$  is then taken to be this representation of the rotational difference between  $\bar{\mathbf{x}}_2$  and  $\bar{\mathbf{m}}_{2,i}$ , such that  $\mathbf{d}_i$  is given in full by

$$\begin{aligned} \mathbf{d}_{1,i} &= \mathbf{x}_1 - \mathbf{m}_{1,i} \\ \mathbf{d}_{2,i} &= 2 \text{vec} \left( \bar{\mathbf{x}}_2 \otimes \bar{\mathbf{m}}_{2,i}^{-1} \right). \end{aligned}$$

By the definition of  $\mathbf{d}_i$ , the Gaussian pdf can be reformulated as

$$p_g(\mathbf{x}; \mathbf{m}_i, \mathbf{P}_i) = |2\pi\mathbf{P}_i|^{-1/2} \exp \left\{ -\frac{1}{2} \mathbf{d}_i^T \mathbf{P}_i^{-1} \mathbf{d}_i \right\},$$

which enables computation of the GMM pdf via Eq. (A.5).

### A.3 Splitting a Multivariate Gaussian Distribution

One of the central algorithms utilized in the splitting Gaussian mixture unscented Kalman filter (SGMUKF) is that of splitting a multivariate Gaussian distribution, which is summarized as: given a component of a GMM pdf represent by weight, mean, and covariance  $w$ ,  $\mathbf{m}$ , and  $\mathbf{P}$  respectively, a set of new components

is generated by splitting along the  $k^{\text{th}}$  column of the square-root matrix,  $\mathbf{S}$ , of the covariance matrix, and so the splitting procedure follows as

$$w_i = \tilde{w}_i w \quad (\text{A.7a})$$

$$\mathbf{m}_i = \mathbf{m} + \tilde{m}_i \mathbf{s}_k \quad (\text{A.7b})$$

$$\mathbf{P}_i = \mathbf{S}_i \mathbf{S}_i^T, \quad (\text{A.7c})$$

where  $\mathbf{s}_k$  is the  $k^{\text{th}}$  column of the square-root factor,  $\mathbf{S}$ , and  $\mathbf{S}_i$  is the square-root factor of the  $i^{\text{th}}$  new component, which is

$$\mathbf{S}_i = [\mathbf{s}_1, \dots, \tilde{\sigma}_i \mathbf{s}_k, \dots, \mathbf{s}_n].$$

Furthermore,  $\tilde{w}_i$ ,  $\tilde{m}_i$ , and  $\tilde{\sigma}_i$ , are dictated by the univariate splitting library chosen for the multivariate splitting process. Since the algorithm relies on adding an offset to the original component mean, some adaptation for the case where a quaternion is present in the state must be considered. As such, let the original mean,  $\mathbf{m}$  be broken into its non-quaternion and quaternion parts as

$$\mathbf{m} = \begin{bmatrix} \mathbf{m}_1 \\ \bar{\mathbf{m}}_2 \end{bmatrix}.$$

Similarly, let the  $k^{\text{th}}$  column of the square-root factor be broken into  $\mathbf{s}_{1,k}$  and  $\mathbf{s}_{2,k}$  as

$$\mathbf{s}_k = \begin{bmatrix} \mathbf{s}_{1,k} \\ \mathbf{s}_{2,k} \end{bmatrix},$$

where  $\mathbf{s}_{1,k}$  corresponds to the non-quaternion part and  $\mathbf{s}_{2,k}$  corresponds to the quaternion part. Note, however, that  $\mathbf{s}_{2,k}$  is not a quaternion. Since the uncertainty for attitude is typically represented in a lower-dimensional space of small angles,  $\mathbf{s}_{2,k}$

actually represents the portion of  $\mathbf{s}_k$  corresponding to this small-angle space. Then, if the  $i^{\text{th}}$  new component mean is given by non-quaternion and quaternion parts as

$$\mathbf{m}_i = \begin{bmatrix} \mathbf{m}_{1,i} \\ \bar{\mathbf{m}}_{2,i} \end{bmatrix},$$

it is readily observed that by Eq. (A.7b),  $\mathbf{m}_{1,i}$  is

$$\mathbf{m}_{1,i} = \mathbf{m}_1 + \tilde{m}_i \mathbf{s}_k. \quad (\text{A.8})$$

The  $\bar{\mathbf{m}}_{2,i}$  term is found by considering the rotational addition of  $\tilde{m}_i \mathbf{s}_{2,k}$  to the quaternion  $\bar{\mathbf{m}}_2$ . Since  $\tilde{m}_i \mathbf{s}_{2,k}$  is a vector of small angles, this rotational additional is accomplished via quaternion multiplication as

$$\bar{\mathbf{m}}_{2,i} = \begin{bmatrix} \frac{1}{2} \tilde{m}_i \mathbf{s}_{2,k} \\ 1 \end{bmatrix} \otimes \bar{\mathbf{m}}_2. \quad (\text{A.9})$$

Then, since  $\bar{\mathbf{m}}_{2,i}$  will not in general be unit norm, the resultant quaternion is normalized, yielding

$$\bar{\mathbf{m}}_{2,i} = \frac{\bar{\mathbf{m}}_{2,i}}{\|\bar{\mathbf{m}}_{2,i}\|}. \quad (\text{A.10})$$

Splitting of the weight and covariance does not require any modification since no quaternion operations are employed, which is readily observed from Eqs. (A.7a) and (A.7c). Therefore, only the splitting of the mean requires modification, and from the preceding developments, the  $i^{\text{th}}$  new component mean is given by Eqs. (A.8) and (A.9), with the additional need to ensure the unit norm requirement via Eq. (A.10).

## A.4 The Method of Moments

Recall the method of moments approach to combining components of a GMM pdf, which was presented in Section 2.4.1, and is given by

$$w_m = \sum_{k=1}^n w_k \quad (\text{A.11a})$$

$$\mathbf{m}_m = \sum_{k=1}^n \frac{w_k}{w_m} \mathbf{m}_k \quad (\text{A.11b})$$

$$\mathbf{P}_m = \sum_{k=1}^n \frac{w_k}{w_m} (\mathbf{P}_k + \mathbf{m}_k \mathbf{m}_k^T) - \mathbf{m}_m \mathbf{m}_m^T. \quad (\text{A.11c})$$

To adapt the method of moments approach for the inclusion of attitude quaternions, first note that the equations for the method of moments may be rewritten as

$$w_m = \sum_{k=1}^n w_k \quad (\text{A.12a})$$

$$\mathbf{m}_m = \sum_{k=1}^n \frac{w_k}{w_m} \mathbf{m}_k \quad (\text{A.12b})$$

$$\mathbf{P}_m = \sum_{k=1}^n \frac{w_k}{w_m} [\mathbf{P}_k + (\mathbf{m}_k - \mathbf{m}_m)(\mathbf{m}_k - \mathbf{m}_m)^T], \quad (\text{A.12c})$$

where it is seen that the only term changed in the new version is that of  $\mathbf{P}_m$ . The equality of Eq. (A.12c) to Eq. (A.11c) is established by starting from Eq. (A.12c) and expanding the outer product of the term  $(\mathbf{m}_k - \mathbf{m}_m)$  with itself, which yields

$$\mathbf{P}_m = \sum_{k=1}^n \frac{w_k}{w_m} (\mathbf{P}_k + \mathbf{m}_k \mathbf{m}_k^T - \mathbf{m}_m \mathbf{m}_k^T - \mathbf{m}_k \mathbf{m}_m^T + \mathbf{m}_m \mathbf{m}_m^T).$$

Then, by breaking the summation into parts and using Eq. (A.11b), it is found that

$$\mathbf{P}_m = \sum_{k=1}^n \frac{w_k}{w_m} (\mathbf{P}_k + \mathbf{m}_k \mathbf{m}_k^T) - 2\mathbf{m}_m \mathbf{m}_m^T + \sum_{k=1}^n \frac{w_k}{w_m} \mathbf{m}_m \mathbf{m}_m^T. \quad (\text{A.13})$$

Since, from Eq. (A.11a),  $w_m = \sum_{k=1}^n w_k$ , it follows that  $\sum_{k=1}^n \frac{w_k}{w_m} = 1$ , such that Eq. (A.13) becomes

$$\mathbf{P}_m = \sum_{k=1}^n \frac{w_k}{w_m} (\mathbf{P}_k + \mathbf{m}_k \mathbf{m}_k^T) - \mathbf{m}_m \mathbf{m}_m^T,$$

and the desired result that Eq. (A.12c) is equivalent to Eq. (A.11c) is established.

With Eqs. (A.12) as the relevant equations for the method of moments, we now consider the situation in which an attitude quaternion is present. In this case, the implementation of Eq. (A.12a) poses no difficulty and is left unchanged. However, since Eq. (A.12b) requires the averaging of the component means, the approach must be adapted for the presence of quaternions. As such, let each of the  $n$  means,  $\mathbf{m}_k$  be broken down into a non-quaternion part and a quaternion part as

$$\mathbf{m}_k = \begin{bmatrix} \mathbf{m}_{1,k} \\ \bar{\mathbf{m}}_{2,k} \end{bmatrix}.$$

Similarly, let the merged mean  $\mathbf{m}_m$  be broken down into a non-quaternion part and a quaternion part as

$$\mathbf{m}_m = \begin{bmatrix} \mathbf{m}_{1,m} \\ \bar{\mathbf{m}}_{2,m} \end{bmatrix}.$$

Then,  $\mathbf{m}_{1,m}$  can be determined by the standard approach of Eq. (A.12b), i.e.

$$\mathbf{m}_{1,m} = \sum_{k=1}^n \frac{w_k}{w_m} \mathbf{m}_{1,k}.$$

The quaternion part of  $\mathbf{m}_m$  is simply an average quaternion. Therefore,  $\bar{\mathbf{m}}_{2,m}$  can be found using the approach of Section A.1 with the  $k^{\text{th}}$  quaternion in the averaging process being given by  $\bar{\mathbf{m}}_{2,k}$  and an associated weight  $\frac{w_k}{w_m}$ .

To determine the merged covariance  $\mathbf{P}_m$ , define the difference from the  $k^{\text{th}}$  mean to the merged mean to be  $\mathbf{d}_k = \mathbf{m}_k - \mathbf{m}_m$ . Again, separate  $\mathbf{d}_k$  into a non-quaternion part and a quaternion part as

$$\mathbf{d}_k = \begin{bmatrix} \mathbf{d}_{1,k} \\ \bar{\mathbf{d}}_{2,k} \end{bmatrix}.$$

The non-quaternion part of  $\mathbf{d}_k$  is readily found from simple vector subtraction, and the quaternion part of  $\mathbf{d}_k$  can be defined as twice the vector part of  $\bar{\mathbf{m}}_{2,k} \otimes \bar{\mathbf{m}}_{2,m}^{-1}$ , yielding  $\mathbf{d}_k$  as

$$\mathbf{d}_k = \begin{bmatrix} \mathbf{m}_{1,k} - \mathbf{m}_{1,m} \\ 2 \text{vec}(\bar{\mathbf{m}}_{2,k} \otimes \bar{\mathbf{m}}_{2,m}^{-1}) \end{bmatrix}.$$

Then, by substituting the definition of  $\mathbf{d}_k$  into Eq. (A.12c), the merged covariance is found by

$$\mathbf{P}_m = \sum_{k=1}^n \frac{w_k}{w_m} (\mathbf{P}_k + \mathbf{d}_k \mathbf{d}_k^T).$$

## A.5 The Kalman Filter Update

The next issue to be addressed when utilizing quaternions is that of the update under the Kalman filter paradigm. Recall that the state update is given by

$$\hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k^-),$$

where  $\hat{\mathbf{x}}_k^+$  is the *a posteriori* state estimate,  $\hat{\mathbf{x}}_k^-$  is the *a priori* state estimate,  $\mathbf{K}_k$  is the Kalman gain,  $\mathbf{y}_k$  is the incoming measurement data, and  $\hat{\mathbf{y}}_k^-$  is the filter's prediction of the measurement using the *a priori* state distribution. Let the *a priori*



and *a posteriori* state estimates be broken in non-quaternion and quaternion parts as

$$\hat{\mathbf{x}}_k^- = \begin{bmatrix} \hat{\mathbf{x}}_{1,k}^- \\ \hat{\mathbf{x}}_{2,k}^- \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{x}}_k^+ = \begin{bmatrix} \hat{\mathbf{x}}_{1,k}^+ \\ \hat{\mathbf{x}}_{2,k}^+ \end{bmatrix}.$$

Furthermore, let  $\Delta \mathbf{x}_k$  be defined as

$$\Delta \mathbf{x}_k = \mathbf{K}_k(\mathbf{y}_k - \hat{\mathbf{y}}_k^-),$$

such that it too may be broken into parts as

$$\Delta \mathbf{x}_k = \begin{bmatrix} \Delta \mathbf{x}_{1,k} \\ \Delta \mathbf{x}_{2,k} \end{bmatrix}.$$

Similar to the discussion of the modified splitting algorithm,  $\Delta \mathbf{x}_{2,k}$  represents an attitude update in a small-angle space which is to be added onto the *a priori* quaternion estimate. It is clear that the non-quaternion portion of the update remains unchanged, and therefore

$$\hat{\mathbf{x}}_{1,k}^+ = \hat{\mathbf{x}}_{1,k}^- + \Delta \mathbf{x}_{1,k}.$$

The quaternion portion of the update, however, is found by considering the rotational addition of  $\Delta \mathbf{x}_{2,k}$  to the *a priori* quaternion estimate  $\hat{\mathbf{x}}_k^-$ . Since  $\Delta \mathbf{x}_{2,k}$  is a vector of small angles, this rotational additional is accomplished via quaternion multiplication as

$$\hat{\mathbf{x}}_k^+ = \begin{bmatrix} \frac{1}{2}\Delta \mathbf{x}_{2,k} \\ 1 \end{bmatrix} \otimes \hat{\mathbf{x}}_k^-.$$

Then, since  $\hat{\mathbf{x}}_k^+$  will not in general be unit norm, the resultant quaternion is normalized, yielding

$$\hat{\mathbf{x}}_k^+ = \frac{\hat{\mathbf{x}}_k^+}{\|\hat{\mathbf{x}}_k^+\|}.$$

This ad-hoc re-normalization procedure is in fact equivalent to considering a constrained optimization problem (which accounts for the unit norm constraint on the quaternion) for determining the Kalman update.<sup>76</sup>

## A.6 The Unscented Transform

The final algorithmic alteration for the inclusion of attitude is that of the UT, which was given in its standard form in Section 3.2.1. To adapt the UT for attitude, consider a nonlinear function of the form

$$\mathbf{z} = \mathbf{g}(\mathbf{x}) ,$$

where  $\mathbf{x}$  is described by a known mean and covariance, respectively  $\mathbf{m}_x$  and  $\mathbf{P}_x$ . The UT seeks to approximate the mean and covariance of the output,  $\mathbf{z}$ , which are denoted by  $\mathbf{m}_z$  and  $\mathbf{P}_z$ . To facilitate the inclusion of attitude, let the input  $\mathbf{x}$  and its associated mean  $\mathbf{m}_x$  be given by a non-quaternion part and a quaternion part, such that

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \bar{\mathbf{x}}_2 \end{bmatrix} \quad \text{and} \quad \mathbf{m}_x = \begin{bmatrix} \mathbf{m}_{1,x} \\ \bar{\mathbf{m}}_{2,x} \end{bmatrix} .$$

Accordingly, the set of  $K$  sigma-points associated with the input are also given by a non-quaternion and a quaternion part, yielding

$$\mathbf{x}_i = \begin{bmatrix} \mathbf{x}_{1,i} \\ \bar{\mathbf{x}}_{2,i} \end{bmatrix} ,$$

and the weights associated with the sigma-points are given by  $w_i$  where  $i \in \{1, \dots, K\}$  and  $\sum_{i=1}^K w_i = 1$ . Then, the set of transformed sigma-points are given by

$$\mathbf{z}_i = \mathbf{g}(\mathbf{x}_i) \quad \forall \quad i \in \{1, \dots, K\} ,$$

where, for generality, it is assumed that the transformed sigma-points are comprised of a non-quaternion part and a quaternion part, or

$$\mathbf{z}_i = \begin{bmatrix} \mathbf{z}_{1,i} \\ \bar{\mathbf{z}}_{2,i} \end{bmatrix}.$$

Since, in general, the transformed sigma-points have both non-quaternion and quaternion parts, so then does the transformed mean, such that  $\mathbf{m}_z$  is given by

$$\mathbf{m}_z = \begin{bmatrix} \mathbf{m}_{1,z} \\ \bar{\mathbf{m}}_{2,z} \end{bmatrix}.$$

Recalling from the standard form of the UT that the transformed mean is given by the weighted sum of the transformed sigma-points, it follows then that the non-quaternion part of the mean is

$$\mathbf{m}_{1,z} = \sum_{i=1}^K w_i \mathbf{z}_{1,i}.$$

As previously discussed, the simple implementation of a normalized weighted sum of the quaternion part of the transformed sigma-points no longer provides an average that retains the properties of a quaternion. Therefore,  $\bar{\mathbf{m}}_{2,z}$  is determined by applying the averaging algorithm for quaternions given in Section A.1 with  $\bar{\mathbf{z}}_{2,i}$  as the quaternions to be averaged, and  $w_i$  as their associated weights. Now, let the difference between the transformed sigma-points and the transformed mean be defined by  $\mathbf{d}_{z,i}$ , such that

$$\mathbf{d}_{z,i} = \begin{bmatrix} \mathbf{d}_{1,z,i} \\ \mathbf{d}_{2,z,i} \end{bmatrix}.$$

Since  $\mathbf{d}_{1,z,i}$  involves subtraction of the non-quaternion parts of  $\mathbf{z}$  and  $\mathbf{m}_z$ , no alteration needs to be made. However, to compute  $\mathbf{d}_{2,z,i}$ , consider the rotational difference

of  $\bar{\mathbf{Z}}_{2,i}$  from  $\bar{\mathbf{m}}_{2,z}$  as given by

$$\delta\bar{\mathbf{q}}_i = \bar{\mathbf{Z}}_{2,i} \otimes \bar{\mathbf{m}}_{2,z,i}^{-1}.$$

Then, assuming that  $\delta\bar{\mathbf{q}}_i$  can be well represented by a small angle quaternion yields

$$\delta\bar{\mathbf{q}}_i = \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}_i \\ 1 \end{bmatrix},$$

such that the vector of small rotation angles described by  $\delta\boldsymbol{\theta}_i$  can be computed as twice the vector part of  $\delta\bar{\mathbf{q}}_i$ , or

$$\delta\boldsymbol{\theta}_i = 2 \text{vec} \left( \bar{\mathbf{Z}}_{2,i} \otimes \bar{\mathbf{m}}_{2,z,i}^{-1} \right),$$

where  $\text{vec}(\cdot)$  denotes taking only the vector part of the quaternion argument. The distance  $\mathbf{d}_{2,i}$  is then taken to be this representation of the rotational difference between  $\bar{\mathbf{Z}}_{2,i}$  and  $\bar{\mathbf{m}}_{2,z,i}$ , such that  $\mathbf{d}_i$  is given in full by

$$\begin{aligned} \mathbf{d}_{1,z,i} &= \mathbf{Z}_{1,i} - \mathbf{m}_{1,z} \\ \mathbf{d}_{2,z,i} &= 2 \text{vec} \left( \bar{\mathbf{Z}}_{2,i} \otimes \bar{\mathbf{m}}_{2,z}^{-1} \right). \end{aligned}$$

Then, the covariance of the output of the nonlinear function is found using  $\mathbf{d}_{z,i}$  as

$$\mathbf{P}_z = \sum_{i=1}^K w_i \mathbf{d}_{z,i} \mathbf{d}_{z,i}^T.$$

Additionally, if it desired to compute the cross-covariance between the input and the output, define  $\mathbf{d}_{x,i}$  to be the difference between the sigma-points and their associated mean. Then, let  $\mathbf{d}_{x,i}$  be given by

$$\mathbf{d}_{x,i} = \begin{bmatrix} \mathbf{d}_{1,x,i} \\ \mathbf{d}_{2,x,i} \end{bmatrix}.$$

Following the same process as developed for  $\mathbf{d}_{z,i}$ , it follows that

$$\begin{aligned}\mathbf{d}_{1,x,i} &= \boldsymbol{\mathcal{X}}_{1,i} - \mathbf{m}_{1,x} \\ \mathbf{d}_{2,x,i} &= 2 \operatorname{vec} \left( \bar{\boldsymbol{\mathcal{X}}}_{2,i} \otimes \bar{\mathbf{m}}_{2,x}^{-1} \right) ,\end{aligned}$$

such that the cross-covariance between the input and the output of the nonlinear function is

$$\mathbf{P}_{xz} = \sum_{i=1}^K w_i \mathbf{d}_{x,i} \mathbf{d}_{z,i}^T .$$

As with the standard UT algorithm, the selection of the sigma-points and their associated weights is chosen in accordance with the symmetric sigma-point set. Let the sigma-points, denoted by  $\boldsymbol{\mathcal{X}}_i$  be comprised of a non-quaternion part and a quaternion part as

$$\boldsymbol{\mathcal{X}}_i = \begin{bmatrix} \boldsymbol{\mathcal{X}}_{1,i} \\ \bar{\boldsymbol{\mathcal{X}}}_{2,i} \end{bmatrix} .$$

The standard symmetric sigma-points are found by adding the  $i^{\text{th}}$  scaled column of the square-root factor of  $\mathbf{P}_x$  to the mean,  $\mathbf{m}_x$ . In the case of the non-quaternion part of  $\boldsymbol{\mathcal{X}}_i$ , this process requires no modification, and so the  $K = 2n$  values of  $\boldsymbol{\mathcal{X}}_{1,i}$  and their associated weights are given by

$$\begin{aligned}w_i &= \frac{1}{2n} \\ \boldsymbol{\mathcal{X}}_{1,i} &= \mathbf{m}_{1,x} + \sqrt{n} \mathbf{s}_{1,x,i} \\ w_{i+n} &= \frac{1}{2n} \\ \boldsymbol{\mathcal{X}}_{1,i+n} &= \mathbf{m}_{1,x} - \sqrt{n} \mathbf{s}_{1,x,i} ,\end{aligned}$$

for  $i \in \{1, \dots, n\}$ , where  $n$  is the dimension of the input  $\mathbf{x}$ ,  $\mathbf{S}_x$  is a square-root factor of  $\mathbf{P}_x$  such that  $\mathbf{P}_x = \mathbf{S}_x \mathbf{S}_x^T$ , and  $\mathbf{s}_{x,i}$  is the  $i^{\text{th}}$  column of  $\mathbf{S}_x$ . Furthermore,  $\mathbf{s}_{1,x,i}$  is the portion of the  $i^{\text{th}}$  column of the square-root factor of  $\mathbf{P}_x$  which represents the non-quaternion related elements of the square-root matrix (and consequently covariance matrix), which is given by breaking  $\mathbf{s}_{x,i}$  into parts as

$$\mathbf{s}_{x,i} = \begin{bmatrix} \mathbf{s}_{1,x,i} \\ \mathbf{s}_{2,x,i} \end{bmatrix}.$$

To generate the quaternion part of  $\boldsymbol{\chi}_i$ , we consider the rotational addition (or subtraction) of  $\sqrt{n}\mathbf{s}_{2,x,i}$  to the quaternion part of the mean,  $\bar{\mathbf{m}}_{2,x}$ . Since  $\sqrt{n}\mathbf{s}_{2,x,i}$  is a vector of small angles, this rotational additional is performed using quaternion multiplication, such that the quaternion part of the sigma-points and the associated weights are given by

$$\begin{aligned} w_i &= \frac{1}{2n} \\ \bar{\boldsymbol{\chi}}_{2,i} &= \begin{bmatrix} \frac{1}{2}\sqrt{n}\mathbf{s}_{2,x,i} \\ 1 \end{bmatrix} \otimes \bar{\mathbf{m}}_{2,x} \\ w_{i+n} &= \frac{1}{2n} \\ \bar{\boldsymbol{\chi}}_{2,i+n} &= \begin{bmatrix} -\frac{1}{2}\sqrt{n}\mathbf{s}_{2,x,i} \\ 1 \end{bmatrix} \otimes \bar{\mathbf{m}}_{2,x}. \end{aligned}$$

Then, since the generated quaternion elements of the sigma-points,  $\bar{\boldsymbol{\chi}}_{2,i}$ , will not in general be unit norm, the resultant quaternions are normalized, yielding

$$\bar{\boldsymbol{\chi}}_{2,i} = \frac{\bar{\boldsymbol{\chi}}_{2,i}}{\|\bar{\boldsymbol{\chi}}_{2,i}\|},$$

for  $i \in \{1, \dots, 2n\}$ .

## Bibliography

- [1] Daniel L. Alspach and Harold W. Sorenson. Nonlinear Bayesian estimation using Gaussian sum approximations. *IEEE Transactions in Automatic Control*, AC-17(4):439–448, August 1972.
- [2] Brian D.O. Anderson and John B. Moore. *Optimal Filtering*. Dover, 2005.
- [3] M. Athans and M. C. Schweppe. Gradient matrices and matrix calculations. Technical Note 1965-53, MIT Lincoln Laboratory, November 1965.
- [4] Michael Athans, Richard P. Wishner, and Anthony Bertolini. Suboptimal state estimation for continuous-time nonlinear systems from discrete noisy measurements. *IEEE Transactions on Automatic Control*, AC-13(5):504–514, October 1968.
- [5] N.J. Bachman. SPK required reading. [http://naif.jpl.nasa.gov/pub/naif/toolkit\\_docs/MATLAB/req/spk.html](http://naif.jpl.nasa.gov/pub/naif/toolkit_docs/MATLAB/req/spk.html), March 2010.
- [6] Ayanendranath Basu, Ian R. Harris, Nils L. Hjort, and M. C. Jones. Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85(3):549–559, 1998.
- [7] Gerald J. Bierman. Covariance propagation via its eigenvalues and eigenvectors. *AIAA Journal*, 8(5):958–959, May 1970.

- [8] Minkang Cheng, John C. Ries, and Byron D. Tapley. Solar radiation pressure modeling for GRACE. 37th COSPAR Assembly, 2008.
- [9] Dayne G. Cook. Solar radiation pressure modeling issues for high altitude satellites. Master's thesis, Air Force Institute of Technology, 2001.
- [10] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2nd edition, 2006.
- [11] Kyle J. DeMars, Moriba K. Jah, Daniel R. Giza, and Thomas M. Kelecy. Orbit determination performance improvements for high area-to-mass ratio space object tracking using an adaptive Gaussian mixtures estimation algorithm. Toulouse, France, September 2009. 21st International Symposium on Space Flight Dynamics.
- [12] Daniel R. Giza, Puneet Singla, and Moriba K. Jah. An approach for nonlinear uncertainty propagation: Application to orbital mechanics. Chicago, Illinois, August 2009. AIAA Guidance, Navigation, and Control Conference.
- [13] Jacob Goldberger and Hagai Aronowitz. A distance measure between GMMs based on the unscented transform and its application to speaker recognition. In *Proceedings of Interspeech*, pages 1985–1989, 2005.
- [14] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering: Theory and Practice*. Prentice Hall, 1993.



- [15] Andreas Griewank and Andrea Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, Philadelphia, Pennsylvania, 2nd edition, 2008.
- [16] Uwe D. Hanebeck, Kai Briechle, and Andreas Rauh. Progressive Bayes: A new framework for nonlinear state estimation. In *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*, volume 5099 of *Proceedings of the SPIE*, pages 256–267, 2003.
- [17] Uwe D. Hanebeck and Olga Feiermann. Progressive Bayesian estimation for nonlinear discrete-time systems: The filter step for scalar measurements and multidimensional states. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, volume 5, pages 5366–5371, 2003.
- [18] Marco F. Huber, Tim Bailey, Hugh Durrant-Whyte, and Uwe D. Hanebeck. On entropy approximation for Gaussian mixture random vectors. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 181–188, 2008.
- [19] R. S. Hujsak. Solar pressure. In *Proceedings of the Artificial Satellite Theory Workshop, USNO*, pages 54–72, November 1993.
- [20] Paul J. Huxel. *Navigation Algorithms and Observability Analysis for Formation Flying Missions*. PhD thesis, The University of Texas at Austin, 2006.
- [21] Kazufumi Ito and Kaiqi Xiong. Gaussian filters for nonlinear filtering problems. *IEEE Transactions on Automatic Control*, 45(5):910–927, May 2000.

- [22] Andrew H. Jazwinski. *Stochastic Processes and Filtering Theory*. Dover, 1998.
- [23] Bing Jian and Baba C. Vemuri. A robust algorithm for point set registration using mixture of Gaussians. In *Proceedings of the Tenth IEEE International Conference on Computer Vision*, volume 2, pages 1246–1251, 2005.
- [24] Sergio F. Juárez and William R. Schucany. A note on the asymptotic distribution of the minimum density power divergence estimator. In *Optimality: The Second Erich L. Lehmann Symposium*, volume 49, pages 334–339, 2006.
- [25] Simon Julier and Jeffrey Uhlmann. Unscented filtering and nonlinear estimation. IEEE, March 2004.
- [26] Simon J. Julier, Jeffrey K. Uhlmann, and Hugh F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of the American Control Conference*, volume 3, pages 1628–1632, June 1995.
- [27] John L. Junkins, Maruthi R. Akella, and K. Terry Alfriend. Non-Gaussian error propagation in orbit mechanics. *Journal of the Astronautical Sciences*, 44(4):541–563, 1996.
- [28] Rudolph E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME – Journal of Basic Engineering*, 82:35–45, 1960.
- [29] James E. Keat. Analysis of least squares attitude determination routine doaop. Technical Report CSC/TM-77/6034, Computer Sciences Corporation, February 1977.

- [30] Andrei N. Kolmogorov. On the Shannon theory of information transmission in the case of continuous signals. *IRE Transactions on Information Theory*, pages 102–108, December 1956.
- [31] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, March 1951.
- [32] Kok-Lam Lai and John Crassidis. Second-order divided-difference filter using a generalized complex-step approximation. Hilton Head, South Carolina, August 2007. AIAA Guidance, Navigation and Control Conference and Exhibit.
- [33] Kok-Lam Lai, John Crassidis, Yang Cheng, and Jongrae Kim. New complex-step derivative approximations with application to second-order kalman filtering. San Francisco, California, August 2005. AIAA Guidance, Navigation, and Control Conference and Exhibit.
- [34] B. P. Lathi. *Linear Systems and Signals*. Oxford University Press, 2002.
- [35] William M. Lear. Kalman filtering techniques. Technical Report JSC-20688, NASA, 1985.
- [36] Yan Li and Lei Li. A novel split and merge EM algorithm for Gaussian mixture model. In *Fifth International Conference on Natural Computation*, volume 6, pages 479–483, 2009.
- [37] Yan Li and Lei Li. A split and merge EM algorithm for color image segmentation. In *International Conference on Intelligent Computing and Intelligent Systems*, volume 4, pages 395–399, 2009.

- [38] J.-C. Liou and N. L. Johnson. Risks in space from orbiting debris. *Science*, 311(5759):340–341, January 2006.
- [39] John B. Lundberg and Bob E. Schutz. Recursion formulas of Legendre functions for use with nonsingular geopotential models. *Journal of Guidance, Control, and Dynamics*, 11(1):31–38, January-February 1988.
- [40] Richard H. Lyon. Geosynchronous orbit determination using space surveillance network observations and improved radiative force modeling. Master’s thesis, Massachusetts Institute of Technology, 2004.
- [41] Manoranjan Majji, John L. Junkins, and James D. Turner. A perturbation method for estimation of dynamic systems. *Nonlinear Dynamics*, 60(3):303–325, 2010.
- [42] F. Landis Markley, Yang Cheng, John L. Crassidis, and Yaakov Oshman. Averaging quaternions. *Journal of Guidance, Control, and Dynamics*, 30(4):1193–1196, July-August 2007.
- [43] Peter S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 1. Academic Press, 1979.
- [44] Peter S. Maybeck and Brian D. Smith. Multiple model tracker based on Gaussian mixture reduction for maneuvering targets in clutter. In *Proceedings of the Eighth International Conference on Information Fusion*, pages 40–47, 2005.
- [45] O. Montenbruck and E. Gill. *Satellite Orbits: Models, Methods, and Applications*. Springer-Verlag, Berlin, 2005.

- [46] T. Moyer. Mathematical formulation of the double-precision orbit determination program (dpodp). Technical Report 32-1527, NASA Jet Propulsion Laboratory, 1971.
- [47] T. Moyer. *Formulation for Observed and Computed Values of Deep Space Network Data Types for Navigation*. JPL, 2000.
- [48] Ryan S. Park and Daniel J. Scheeres. Nonlinear semi-analytic methods for trajectory estimation. *Journal of Guidance, Control, and Dynamics*, 30(6):1668–1676, November-December 2007.
- [49] Sang H. Park. *Nonlinear Trajectory Navigation*. PhD thesis, The University of Michigan, 2007.
- [50] Samuel Pines. Uniform representation of the gravitational potential and its derivatives. *AIAA Journal*, 11(11):1508–1511, November 1973.
- [51] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, Cambridge, 3rd edition, 2007.
- [52] Surajit Ray. *Distance-Based Model-Selection with Application to the Analysis of Gene Expression Data*. PhD thesis, The Pennsylvania State University, 2003.
- [53] Alfréd Rényi. On measures of entropy and information. In Jerzy Neyman, editor, *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1 of *Contributions to the Theory of Statistics*, pages 547–561. University of California Press, June-July 1961.

- [54] Ben Rose. Resident space object models. Technical report, Emergent Space Technologies, Inc., 2010.
- [55] Wilson J. Rugh. *Linear System Theory*. Prentice Hall, Upper Saddle River, NJ, 2nd edition, 1996.
- [56] Paul W. Schumacher, Jr. US naval space surveillance upgrade program 1999-2003. Fifth European Conference on Space Debris, March 2009.
- [57] Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948.
- [58] Malcolm D. Shuster. A survey of attitude representations. *The Journal of the Astronautical Sciences*, 41(4):439–517, 1993.
- [59] Gerald L. Smith, Stanley F. Schmidt, and Leonard A. McGee. Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle. Technical Report R-135, NASA, 1962.
- [60] Kai-Sheng Song. Rényi information, loglikelihood and an intrinsic distribution mmeasure. *Journal of Statistical Planning and Inference*, 93(1-2):51–69, February 2001.
- [61] H. W. Sorenson and D. L. Alspach. Recursive Bayesian estimation using Gaussian sums. *Automatica*, 7:465–479, 1971.
- [62] William Squire and George Trapp. Using complex variables to estimate derivatives of real functions. *SIAM Review*, 40(1):110–112, March 1998.

- [63] Hsi Guang Sung. *Gaussian Mixture Regression and Classification*. PhD thesis, Rice University, 2004.
- [64] B. Tapley, J. Ries, S. Bettadpur, D. Chambers, M. Cheng, F. Condi, and S. Poole. The GGM03 mean Earth gravity model from GRACE. *AGU Fall Meeting Abstracts*, December 2007.
- [65] Byron D. Tapley, Bob E. Schutz, and George H. Born. *Statistical Orbit Determination*. Elsevier Academic Press, New York, NY, 2004.
- [66] Gabriel Terejanu, Puneet Singla, Tarunraj Singh, and Peter Scott. Uncertainty propagation for nonlinear dynamic systems using Gaussian mixture models. *Journal of Guidance, Control, and Dynamics*, 31(6):1623–1633, November–December 2008.
- [67] Jeffrey K. Uhlmann. *Simultaneous Map Building and Localization for Real Time Applications*. PhD thesis, University of Oxford, 1994.
- [68] Robert Vallée. Information entropy and state observation of a dynamical system. In B. Bouchon and R. Yager, editors, *Uncertainty in Knowledge-Based Systems*, volume 286 of *Lecture Notes in Computer Science*, pages 403–405. Springer Berlin/Heidelberg, 1987.
- [69] Robert Vallée. Evolution of uncertainty about the state of a dynamical system. *Kybernetes*, 27(9):1007–1011, 1998.

- [70] Rudolph van der Merwe. *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, Oregon Health and Science University, 2004.
- [71] Ba-Ngu Vo and Wing-Kin Ma. The Gaussian mixture probability hypothesis density filter. *IEEE Transactions in Signal Processing*, 54(11):4091–4104, November 2006.
- [72] Jason L. Williams and Peter S. Maybeck. Cost-function-based Gaussian mixture reduction for target tracking. In *Proceedings of the Sixth International Conference of Information Fusion*, pages 1047–1054, 2003.
- [73] James Woodburn and Sergei Tanygin. Detection of non-linearity effects during orbit estimation. San Diego, CA, February 2010. 20th AAS/AIAA Space Flight Mechanics Meeting.
- [74] Zhiying Yao and Dong Liu. Gaussian mixture reduction based on KI divergence. In *2nd International Conference on Future Computer and Communication*, volume 1, pages 630–634, 2010.
- [75] Renato Zanetti, Kyle J. DeMars, and Robert H. Bishop. Underweighting non-linear measurements. *Journal of Guidance, Control, and Dynamics*, 33(5):1670–1675, September-October 2010.
- [76] Renato Zanetti, Manoranjan Majji, Robert H. Bishop, and Daniele Mortari. Norm-constrained Kalman filtering. *Journal of Guidance, Control, and Dynamics*, 32(5):1458–1465, September-October 2009.



- [77] K. Zografos and S. Nadarajah. Expressions for Rényi and Shannon entropies for multivariate distributions. *Statistics & Probability Letters*, 71(1):71–84, January 2005.

## Vita

Kyle Jordan DeMars was born in Texarkana, Texas on June 8, 1983. He began his aerospace engineering studies in the fall of 2001 when he enrolled in The University of Texas at Austin. As an undergraduate student, Kyle began doing research with Dr. Bishop on numerical solutions to libration points in general three-body gravitational systems. Kyle graduated with his Bachelor of Science degree with high honors in the fall of 2004, and his Master of Science in Engineering degree in the spring of 2007. He then continued on at The University of Texas at Austin under Dr. Bishop working on developing algorithms for precision lunar landing navigation, which culminated in two sets of real-time field tests of the developed algorithms. During the summer of 2008 and the summer of 2009, Kyle was selected as a Space Scholar and worked for the Air Force Research Laboratory developing methods for performing autonomous initial orbit determination using angles-only and angle and angle-rate data.

Permanent address: 5400 W. Parmer Ln. Apt. 1035  
Austin, Texas 78727

This dissertation was typeset with  $\text{\LaTeX}^\dagger$  by the author.

---

<sup>†</sup> $\text{\LaTeX}$  is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's  $\text{\TeX}$  Program.